Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

С. Н. Павлов

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Часть 2

Учебное пособие

Томск «Эль Контент» 2011 УДК 004.89(075.8) ББК 32.813я73 П 12

Репензенты:

Сергеев В. Л., докт. техн. наук, проф. кафедры геологии и разработки нефтяных месторождений Томского политехнического университета;

Кориков А. М., проф., зав. кафедрой автоматизированных систем управления ТУСУРа

Павлов С. Н.

П 12 Системы искусственного интеллекта: учеб. пособие. В 2-х частях. / С. Н. Павлов. — Томск: Эль Контент, 2011. — Ч. 2. — 194 с.

ISBN 978-5-4332-0014-2

В учебном пособии рассматриваются теоретические и организационно-методологические вопросы разработки и применения систем искусственного интеллекта. Изложены базовые принципы, подходы, классификация, методы, модели и стратегии систем различного назначения, традиционно считающиеся интеллектуальными: интеллектуальные информационные системы, понимание естественного языка и изображений, представление знаний и обучение, логический вывод и планирование действий. Большое внимание уделяется экспертным системам, обработки естественного языка, машинному зрению.

Пособие предназначено для студентов вузов, обучающихся по специальностям «Компьютерная инженерия», «Программное обеспечение вычислительной техники и автоматизированных систем», а также аспирантов и специалистов, интересующихся вопросами систем искусственного интеллекта.

УДК 004.89(075.8) ББК 32.813я73

Оглавление

Введение				
6	Пла	анирование в интеллектуальных системах	7	
	6.1	Классификация планирования	9	
		6.1.1 Классическое планирование	9	
		6.1.2 Планирование и осуществление действий в реальном мире .	13	
	6.2	Методы решения задач планирования	15	
		6.2.1 Решение задачи методом редукции	15	
		6.2.2 Метод ключевых состояний и ключевых операторов	16	
		6.2.3 Метод анализа средств и целей	17	
		6.2.4 Планирование с помощью логического вывода	20	
	6.3	Примеры планирующих систем	20	
	6.4	История интеллектуального планирования	26	
7	Экс	пертные системы	42	
	7.1	Классификация экспертных систем	42	
	7.2	Экспертные системы первого и второго поколений	47	
	7.3	Классификация ИЭС,		
		взаимосвязь процессов интеграции и гибридизации в ИЭС	48	
	7.4	Структура и компоненты экспертных систем	52	
	7.5	Этапы разработки экспертных систем	55	
	7.6	Представление знаний в экспертных системах	58	
	7.7	Блок (подсистема) объяснений	65	
	7.8	Взаимодействие пользователя с экспертной системой	69	
		7.8.1 Структура системы с естественным языком общения	69	
		7.8.2 Компьютерно-лингвистический подход к диалогу	74	
8	Зна	ния и их представление в интеллектуальных системах	79	
	8.1	Понятие знания, представление знаний	79	
	8.2	Данные и знания в интеллектуальных системах	82	
	8.3	Понятийная структура предметной области	88	
	8.4	Модели знаний	93	
		8.4.1 Логические модели	93	
		8.4.2 Семантические сети	98	
		8.4.3 Фреймы	117	
		8.4.4 Сценарии	123	
9	Сис	темы понимания естественного языка, машинный перевод	130	
	9.1	Искусственный интеллект и компьютерная лингвистика	131	
	9.2	Понимание текстов на естественном языке	132	

4 Оглавление

		9.2.1	Уровни понимания текста интеллектуальной системы	132				
		9.2.2	Уровни интерпретации понимания текста	134				
		9.2.3	Структура языковых форм общения интеллектуального					
			робота	138				
	9.3	Маши	- нный перевод					
		9.3.1	Электронные словари	139				
		9.3.2	Понятие машинного перевода	139				
		9.3.3	Семантические проблемы машинного перевода	140				
		9.3.4	Процесс машинного перевода	142				
		9.3.5	Системы машинного перевода	144				
		9.3.6	Достоинства систем машинного перевода	149				
		9.3.7	Будущее машинного перевода	150				
10	Зрит	гельное	е восприятие мира	154				
	-		внавание образов и обучение	154				
			Зрительное восприятие					
			Основные сведения о распознавании образов					
			Геометрический метод распознавания	160				
			Байесовский метод распознавания	162				
			Синтаксический метод распознавания	164				
	10.2		ютерное зрение	168				
			Типичные задачи компьютерного зрения	169				
			Системы компьютерного зрения	170				
			Этапы понимания изображения					
	10.3		нное зрение	177				
			Обработка визуальной информации	178				
			Система зрения роботов					
3 a:	ключ	ение		185				
Гп	Глоссарий							

ВВЕДЕНИЕ

Соглашения, принятые в книге

Эта пиктограмма означает внимание. Здесь выделена важная информация, требующая акцента на ней. Автор здесь может поделиться с читателем опытом, чтобы помочь избежать некоторых ошибок.

Для улучшения восприятия материала в данной книге используются пикто-



В блоке «На заметку» автор может указать дополнительные сведения или другой взгляд на изучаемый предмет, чтобы помочь читателю лучше понять основные идеи.



Эта пиктограмма означает пример. В данном блоке автор может привести практический пример для пояснения и разбора основных моментов, отраженных в теоретическом материале.

6 Введение

	Выводы
	рамма означает выводы. Здесь автор подводит итоги, обобщает из- ериал или проводит анализ.
?	Контрольные вопросы и задания к главе
	Литература к главе

Глава 6

ПЛАНИРОВАНИЕ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ



Планирование — это область искусственного интеллекта, к которой в настоящее время проявляют значительный интерес.

Одна из причин такой ситуации состоит в том, что в планировании объединяются два основных направления развития искусственного интеллекта — поиск и логика. Это означает, что любой планировщик (комплекс программных средств, предназначенных для поиска планов действий) может рассматриваться либо как программа, в которой осуществляется поиск решения, либо как такая программа, которая (конструктивно) доказывает существование решения. Такое перекрестное обогащение идеями, взятыми из этих двух областей, привело не только к повышению производительности, которое за последнее десятилетие достигло нескольких порядков величины, но и к расширению использования планировщиков в производственных приложениях. К сожалению, еще не сложилось четкое понимание того, какие методы являются наиболее подходящими для задач того или иного типа.

Изучив методы представления и решения задач, мы перейдем к анализу систем решения задач в интеллектуальных системах [1].

Такая система обладает описанием мира и общим механизмом создания целенаправленных стратегий в виде последовательности элементарных операторов, которые бы выполняли разумные действия в окружающем мире.



Целенаправленная стратегия поиска решения задачи называется планом.

Пространством моделей назовем множество всех возможных состояний моделей мира, описываемых алгоритмической системой объекта.

Моделью элементарного действия системы в мире или оператором называется некоторое отношение, определенное в пространстве моделей.

В соответствии с такими определениями дадим вербальное определение плана.

Планом называется помеченный направленный граф, удовлетворяющий следующим требованиям:

- каждая дуга графа помечена оператором, в общем случае оператором-схемой, т. е. семейством операторов, определенных некоторым параметром;
- с каждой вершиной в графе связана некоторая формула, определяющая, в свою очередь, подмножество множества моделей;
- из одной вершины исходят одинаково помеченные дуги;
- подмножество моделей, соответствующее вершине, содержится в области определения оператора, помечающего дуги, исходящие из этой вершины.

Обобщенным планом называется множество планов, которое можно получить из исходного конкретной подстановкой значений параметров.

Простым планом называется план, если для всех вершин оператор является однозначной функцией модели (из каждой вершины исходит ровно одна дуга).

План называется сложным, если каждому результату применения оператора приписывается некоторая оценка правдоподобия его появления или оценка его полезности с точки зрения достижения цели (оценки приписываются дугам, исходящим из вершины).

План называется составным в случае, если альтернативным результатом применения оператора не приписаны оценки или приписаны одинаковые оценки.

Алгоритмическая система, осуществляющая построение планов решения задачи, называется планирующей.

В связи с проблемой планирования и выполнения действий встает задача обобщения моделей, найденных планов и использования их в последующих процессах построения планов и при выполнении действий.

Введение операторов-схем — это один из шагов обобщения моделей. Далее происходит постепенное наращивание множества операторов путем их укрупнения, а также соответствующее обобщение условий их применения, с тем, чтобы создать иерархию планов — от укрупненных к более детализированным.

Мы будем рассматривать два класса миров (сред). Если рассматривать лишь такие варианты среды, которые являются полностью наблюдаемыми, детерминированными, конечными, статическими и дискретными (с точки зрения времени, действий, объектов и результатов), то такая среда называется средой классического планирования. Планировщики, применяемые в реальном мире для решения таких задач, как планирование наблюдений с помощью космического телескопа Хаббл, управление предприятиями, а также осуществление поставок для военных компаний, являются более сложными. Они превосходят свои более простые аналоги и с точки зрения языка представления, и с точки зрения того способа, который применяется в планировщике для взаимодействия с его средой [3, 5–8, 9].

6.1 Классификация планирования

6.1.1 Классическое планирование

Планирование с помощью поиска в пространстве состояний. Наиболее простой алгоритм планирования состоит в использовании поиска в пространстве состояний. Поскольку описания действий в задаче планирования определяют и предусловия, и результаты, существует возможность организовать поиск в обоих направлениях: либо в прямом, от начального состояния, либо в обратном, от цели.

Представление задач в пространстве состояний предполагает задание ряда описаний: состояний, множества операторов и их воздействий на переходы между состояниями, целевых состояний. Описания состояний могут представлять собой строки символов, векторы, двумерные массивы, деревья, списки и т. п. Операторы переводят одно состояние в другое. Иногда они представляются в виде продукций $A \Rightarrow B$, означающих, что состояние A преобразуется в состояние B.

Пространство состояний можно представить как граф, вершины которого помечены состояниями, а дуги — операторами. Если некоторая дуга направлена от вершины n_i к вершине n_j , то вершина n_i называется дочерней, а n_i родительской.

Последовательность вершин $n_{i1}, n_{i2}, \ldots, n_{ik}$, в которой каждая вершина, кроме первой, является дочерней для предыдущей, называется путем длиной k от вершины n_{i1} к n_{ik} .

Таким образом, проблема поиска решения задачи $\langle A, B \rangle$ при планировании по состояниям представляется как проблема поиска на графе пути из A в B. Обычно графы не задаются, а генерируются по мере надобности (неявное задание графа).

Различаются слепые и направленные методы поиска пути. Слепой метод имеет два вида: поиск вглубь и поиск вширь. При поиске вглубь каждая альтернатива исследуется до конца, без учета остальных альтернатив. Метод плох для «высоких» деревьев, так как можно легко проскользнуть мимо нужной ветви и затратить много усилий на исследование «пустых» альтернатив. При поиске вширь на фиксированном уровне исследуются все альтернативы и только после этого осуществляется переход на следующий уровень. Метод может оказаться хуже метода поиска вглубь, если в графе все пути, ведущие к целевой вершине, расположены примерно на одной и той же глубине. Оба слепых метода требуют большой затраты времени, и поэтому необходимы направленные методы поиска.

Метод ветвей и грании. Из формирующихся в процессе поиска неоконченных путей выбирается самый короткий и продлевается на один шаг. Полученные новые неоконченные пути (их столько, сколько ветвей в данной вершине) рассматриваются наряду со старыми, и вновь продлевается на один шаг кратчайший из них. Процесс повторяется до первого достижения целевой вершины, решение запоминается. Затем из оставшихся неоконченных путей исключаются более длинные, чем законченный путь, или равные ему, а оставшиеся продлеваются по такому же алгоритму до тех пор, пока их длина меньше законченного пути. В итоге либо все неоконченные пути исключаются, либо среди них формируется законченный путь, более короткий, чем ранее полученный. Последний путь начинает играть роль эталона и т. д.

Алгоритм кратчайших путей Мура. Исходная вершина x_0 помечается

числом 0. Пусть в ходе работы алгоритма на текущем шаге получено множество дочерних вершин $\Gamma(x_i)$ вершины x_i . Тогда из него вычеркиваются все ранее полученные вершины, оставшиеся помечаются меткой, увеличенной на единицу по сравнению с меткой вершины x_i , и от них проводятся указатели к x_i . Далее, на множестве помеченных вершин, еще не фигурирующих в качестве адресов указателей, выбирается вершина с наименьшей меткой и для нее строятся дочерние вершины. Разметка вершин повторяется до тех пор, пока не будет получена целевая вершина.

Алгоритм Дейкстры определения путей с минимальной стоимостью является обобщением алгоритма Мура за счет введения дуг переменной длины.

Алгоритм Дорана и Мичи поиска с низкой стоимостью. Используется, когда стоимость поиска велика по сравнению со стоимостью оптимального решения. В этом случае вместо выбора вершин, наименее удаленных от начала, как в алгоритмах Мура и Дейкстры, выбирается вершина, для которой эвристическая оценка расстояния до цели наименьшая. При хорошей оценке можно быстро получить решение, но нет гарантии, что путь будем минимальным.

Алгоритм Харта, Нильсона и Рафаэля. В алгоритме объединены оба критерия: стоимость пути до вершины g(x) и стоимость пути от вершины h(x) — в аддитивной оценочной функции f(x) = g(x) + h(x). При условии $h(x) < h_p(x)$, где $h_p(x)$ — действительное расстояние до цели, алгоритм гарантирует нахождение оптимального пути.

Алгоритмы поиска пути на графе различаются также направлением поиска. Существуют прямые, обратные и двунаправленные методы поиска. Прямой поиск идет от исходного состояния и, как правило, используется тогда, когда целевое состояние задано неявно. Обратный поиск идет от целевого состояния и используется тогда, когда исходное состояние задано неявно, а целевое явно. Двунаправленный поиск требует удовлетворительного решения двух проблем: смены направления поиска и оптимизации «точки встречи». Одним из критериев для решения первой проблемы является сравнение «ширины» поиска в обоих направлениях — выбирается то направление, которое сужает поиск. Вторая проблема вызвана тем, что прямой и обратный пути могут разойтись и чем уже поиск, тем это более вероятно.

Прямой и обратный поиск в пространстве состояний представляют собой особые формы поиска полностью упорядоченного плана. В них рассматриваются только строго линейные последовательности действий, непосредственно связанные с начальным или целевым состоянием. Это означает, что такие методы поиска не позволяют воспользоваться преимуществами декомпозиции задачи. Вместо того чтобы обеспечить отдельную проработку каждой подзадачи, эти методы вынуждены всегда поддерживать принятие решений о том, как упорядочить действия, относящиеся ко всем подзадачам.

Планирование с частичным упорядочиванием. Обычно более предпочтительным является планирование, позволяющее работать над несколькими подцелями независимо, достигая их с помощью нескольких субпланов, а затем объединять эти субпланы.

Подобный подход обладает также тем преимуществом, что позволяет добиться большей гибкости при определении последовательности, в которой составляется окончательный план. Это означает, что планировщик вначале может работать над «очевидными» или «важными» решениями, не будучи вынужденным прорабаты-

вать все этапы в хронологическом порядке. Общая стратегия, в которой в процессе поиска выбор определенных этапов откладывается на более позднее время, называется стратегией с наименьшим вкладом. Формального определения стратегии с наименьшим вкладом не существует, поскольку очевидно, что на любом этапе поиска должен быть сделан определенный вклад в окончательное решение, так как в противном случае поиск окажется непродуктивным. Любой алгоритм планирования, способный включить в план два действия без указания того, какое из них должно быть выполнено первым, называется планировщиком с частичным упорядочением.

Планирование с частичным упорядочением может быть реализовано в виде поиска в пространстве планов с частичным упорядочением. Это означает, что поиск начинается с пустого плана. После этого рассматриваются способы уточнения плана до тех пор, пока не удастся составить полный план, который решает данную задачу. Действия, рассматриваемые в этом поиске, являются не действиями в мире, а действиями в планах: добавление в план этапа; наложение упорядочения, согласно которому одно действие должно занять место перед другим.

В отличие от планирования с полным упорядочением, планирование с частичным упорядочением обладает явным преимуществом, поскольку позволяет выполнять декомпозицию задачи на подзадачи. Оно имеет также определенный недостаток, который заключается в том, что состояния не определены явно, поэтому труднее оценить, насколько далек план с частичным упорядочением от достижения цели. К тому же в настоящее время существует гораздо меньшее понимание того, как следует вычислять точные эвристики (прием решения задачи, основанный не на строгих математических моделях и алгоритмах, а на соображениях, восходящих к «здравому смыслу»; как правило, эвристика отражает особенности того, как такие задачи решает человек, когда он не пользуется строго формальными приемами) для планирования с частичным упорядочением, чем для планирования с полным упорядочением.

Наиболее очевидная эвристика состоит в подсчете количества различных открытых предусловий. Такая эвристика может быть улучшена путем вычитания из указанной величины количества открытых предусловий, которые согласуются с литералами в состоянии Start. Как и в случае с полным упорядочением, такая эвристика приводит к переоценке стоимости, если имеются действия, достигающие нескольких подцелей, и недооценке стоимости, если возникают отрицательные взаимодействия этапов плана.

Эвристическая функция используется для выбора плана, подлежащего уточнению. При наличии такого выбора алгоритм вырабатывает преемников на основе определения единственного открытого предусловия, которое следует дополнительно проработать.

Планирование с помощью пропозициональной логики. Планирование может осуществляться по принципу доказательства некоторой теоремы в рамках пропозиционного исчисления. В подобной теореме утверждается, что при наличии начального состояния и аксиом состояния-приемника, которые описывают результаты действий, цель будет истинной в ситуации, которая является результатом некоторой последовательности действий. В ранний период развития искусственного интеллекта данный подход считался слишком неэффективным для того, чтобы

с его помощью можно было находить интересные планы. Проведенные в последнее время разработки в области эффективных алгоритмов формирования рассуждений для пропозициональной логики привели к возрождению интереса к планированию с помощью логических рассуждений.

Основным недостатком описанного пропозиционального подхода являются колоссальные размеры пропозициональной базы знаний, которая формируется на основе первоначальной задачи планирования. Поскольку количество символов действий экспоненциально зависит от арности схемы действий, одним из способов преодоления указанного недостатка может оказаться попытка уменьшить арность. Это можно сделать, заимствовав одну идею из области семантических сетей. В семантических сетях используются только бинарные предикаты; предикаты с большим количеством параметров сводятся к множеству бинарных предикатов, которые описывают каждый параметр отдельно. Этот процесс, называемый расщеплением символов, позволяет устранить необходимость в использовании экспоненциального количества символов.

Расщепление символов само по себе позволяет сократить количество символов, но не приводит к автоматическому уменьшению количества аксиом в базе знаний. Это означает, что если бы каждый символ действия в каждом выражении был просто заменен конъюнкцией трех символов, то общий размер базы знаний остался бы примерно тем же самым. Расщепление символов фактически приводит к уменьшению базы знаний потому, что некоторые из расщепленных символов станут нерелевантными для определенных аксиом и могут быть удалены.

Планировщики, основанные на проверке выполнимости, способны обрабатывать крупные задачи планирования, например, находить оптимальные тридцатиэтапные решения для задач планирования в мире блоков с десятками блоков. Размер пропозиционального представления и стоимость решения в высшей степени зависят от задачи, но в большинстве случаев узким местом становится нехватка памяти, требуемой для хранения пропозициональных аксиом. Одним из интересных результатов этих исследований оказалось то, что алгоритмы поиска с возвратами, такие, как DPLL, часто лучше решают задачи планирования по сравнению с алгоритмами локального поиска, подобными WalkSAT. Это связано с тем, что основная часть пропозициональных аксиом представляет собой хорновские выражения, которые эффективно обрабатываются с помощью метода распространения единичных выражений. Это наблюдение привело к разработке гибридных алгоритмов, в которых комбинируется некоторый метод случайного поиска с возвратами и метод распространения единичных выражений.

Планирование иерархической сети задач. Одной из наиболее привлекательных идей в области решения сложных задач является иерархическая декомпозиция. Основное преимущество иерархической структуры состоит в том, что на каждом уровне иерархии вычислительная задача, военная операция или административная функция сводится к небольшому количеству действий, выполняемых на более низком уровне, поэтому вычислительная стоимость поиска правильного способа упорядочения этих действий для решения текущей задачи очень невелика. С другой стороны, в иерархических методах задача сводится к большому количеству отдельных действий; при решении крупномасштабных задач такой подход становится полностью неприменимым. Но в случае, когда высокоуровневые решения

всегда сводятся к решениям, имеющие низкоуровневые реализации иерархические методы могут привести к созданию алгоритмов планирования с линейными затратами времени.

В методе планирования, основанном на иерархических сетях задач или сетях HTN (Hierarchical Task Network), первоначальный план, который описывает задачу, рассматривается как описание на очень высоком уровне того, что должно быть сделано. Планы уточняются путем применения декомпозиции действий. В каждой декомпозиции действия одно действие высокого уровня сводится к частично упорядоченному множеству действий низкого уровня. Поэтому в декомпозициях действий отражены знания о том, как осуществляются действия. Процесс декомпозиции продолжается до тех пор, пока в плане не останутся только примитивные действия.

В «чистом» планировании HTN планы разрабатываются только путем последовательной декомпозиции действий. Поэтому планирование HTN может рассматриваться как процесс конкретизации описания некоторой деятельности, а не как процесс создания описания деятельности, начиная с пустого действия (как в случае планирования в пространстве состояний и планирования с частичным упорядочением). Однако для некоторых задач (особенно для тех, в которых применяется так называемая «новаторская» постановка с конъюнктивными целями) подход с использованием чистого планирования HTN становится не совсем естественным. Поэтому предпочитают применять гибридный подход, в котором декомпозиция действий используется как уточнение плана в планировании с частичным упорядочением.

6.1.2 Планирование и осуществление действий в реальном мире

Возможность получения полной и правильной информации зависит от того, какой степенью недетерминированности характеризуется мир. При ограниченной недетерминированности действия могут иметь непредсказуемые результаты, но все возможные результаты можно перечислить в аксиомах описания действия. Интеллектуальная система получает способность справляться с ограниченной недетерминированностью, составляя планы, применимые во всех возможных обстоятельствах. При неограниченной недетерминированности множество возможных предусловий или результатов либо неизвестно, либо слишком велико для того, чтобы в нем можно было выполнить полный поиск. Интеллектуальной системе удастся справиться с неограниченной недетерминированностью, только если она способна пересматривать свои планы и/или свою базу знаний.

Существуют четыре описанных ниже метода планирования для осуществления действий в условиях недетерминированности. Первые два из них применимы для ограниченной недетерминированности, а последние два — для неограниченной недетерминированности [8, 9].

Планирование без использования датичков. Этот метод, называемый также совместимым планированием, предусматривает создание стандартных, последовательных планов, которые должны выполняться без учета результатов восприятия. Алгоритм планирования без использования датчиков должен обеспечивать, чтобы

цель достигалась в плане при всех возможных обстоятельствах, независимо от истинного начального состояния и фактических результатов действий. Планирование без использования датчиков основано на идее принуждения, согласно которой мир может быть принудительно переведен в данное конкретное состояние, даже если агент обладает лишь частичной информацией о текущем состоянии. Принуждение не всегда возможно, поэтому планирование без использования датчиков часто является неприменимым.

Контроль выполнения и перепланирование. В данном случае агент, контролирующий выполнение, проверяет свои восприятия для определения того, все ли идет в соответствии с планом. Проблема заключается в неограниченной недетерминированности — всегда могут возникнуть непредвиденные обстоятельства, для которых описания действий, подготовленные агентом, будут неправильными. Поэтому в реальных вариантах среды всегда требуется контроль выполнения. Существует два основных способа организации контроля выполнения: простая, но слабая форма, называемая контролем действий, в которой агент проверяет среду для определения того, что следующее действие окажется применимым; и более сложная, но и более эффективная форма, называемая контролем плана, в которой агент проверяет весь оставшийся план.

Перепланирующий агент знает, что делать, когда происходит что-то непредвиденное: снова вызвать планировщик, чтобы он предложил ему новый план достижения цели. Для предотвращения использования слишком больших затрат времени на планирование такая операция обычно осуществляется в виде попытки исправить старый план — найти способ перехода из текущего непредвиденного состояния обратно в одно из тех состояний, которые были предусмотрены в плане.

При этом подходе агент может использовать любой из описанных выше методов планирования (классический, без использования датчиков или условный) для формирования плана, но использует также контроль выполнения для оценки того, предусмотрена ли в плане фактически сложившаяся текущая ситуация или план должен быть пересмотрен. Перепланирование осуществляется, если что-то происходит не так, как надо. Благодаря этому интеллектуальная система приобретает способность справляться с неограниченной недетерминированностью.

Условное планирование. В этом подходе, называемом также планированием с учетом непредвиденных ситуаций, действия в условиях ограниченной недетерминированности осуществляются путем создания условного плана с различными ответвлениями для самых разных непредвиденных ситуаций, какие только могут возникнуть. Так же как и в классическом планировании, система вначале составляет план, а затем выполняет подготовленный план. Интеллектуальная система определяет, какая часть плана должна быть выполнена, включив в план действия по восприятию для проверки соответствующих условий.

Условное планирование представляет собой один из способов учета неопределенности путем проверки того, что фактически происходит в среде при выполнении заранее заданных пунктов плана. Полная наблюдаемость означает, что агент всегда знает текущее состояние. Но если среда является недетерминированной, то агент не будет способен предвидеть результат своих действий. Агент, занимающийся условным планированием, преодолевает такую недетерминированность, встраивая в свой план (на этапе планирования) условные этапы, в которых прове-

ряется состояние среды (на этапе выполнения), для определения того, что делать дальше. Поэтому проблема состоит в том, как создавать такие условные планы.

В реальном мире очень часто встречается частичная наблюдаемость. В начальном состоянии частично наблюдаемой задачи планирования агент обладает лишь некоторым объемом знаний о действительном состоянии. Простейший способ промоделировать такую ситуацию состоит в том, чтобы принять предположение, что начальное состояние принадлежит к множеству состояний; множество состояний представляет собой способ описания начального доверительного состояния агента (если среда не является полностью наблюдаемой, то агент должен рассуждать о том, в какое множество состояний он может попасть; каждое такое множество называется доверительным состоянием).

Непрерывное планирование. Все планировщики, рассматривавшиеся до сих пор, спроектированы так, что они достигают цели, а затем останавливаются, а непрерывный планировщик предназначен для того, чтобы заниматься планированием в течение всего срока своего существования. Он способен справляться с непредвиденными ситуациями в своей среде, даже если они возникают в ходе того, как агент занимается составлением плана. Он способен также обеспечить отказ от целей и создание дополнительных целей с помощью формулировки цели.

6.2 Методы решения задач планирования

Практически все задачи построения плана действий (решения задач) можно разбить на два типа, которым соответствуют различные модели: планирование в пространстве состояний (SS-проблема) и планирование в пространстве задач (PR-проблема) [1, 4, 5, 8, 9].

Различают многочисленные методы решения задачи при планировании по состояниям (см. 6.1.1).

6.2.1 Решение задачи методом редукции

Этот метод приводит к хорошим результатам потому, что часто решение задач имеет иерархическую структуру. Однако не обязательно требовать, чтобы основная задача и все ее подзадачи решались одинаковыми методами. Редукция полезна для представления глобальных аспектов задачи, а при решении более специфичных задач предпочтителен метод планирования по состояниям. Метод планирования по состояниям можно рассматривать как частный случай метода планирования с помощью редукций, ибо каждое применение оператора в пространстве состояний означает сведение исходной задачи к двум более простым, из которых одна является элементарной. В общем случае редукция исходной задачи не сводится к формированию таких двух подзадач, из которых хотя бы одна была элементарной.

Поиск планирования в пространстве задач заключается в последовательном сведении исходной задачи к все более простым до тех пор, пока не будут получены только элементарные задачи. Частично упорядоченная совокупность таких задач составит решение исходной задачи. Расчленение задачи на альтернативные множества подзадач удобно представлять в виде И/ИЛИ-графа. В таком графе всякая вершина, кроме концевой, имеет либо конъюнктивно связанные дочерние вершины

(И-вершина), либо дизъюнктивно связанные (ИЛИ-вершина). В частном случае, при отсутствии И-вершин, имеет место граф пространства состояний. Концевые вершины являются либо заключительными (им соответствуют элементарные задачи), либо тупиковыми. Начальная вершина (корень И/ИЛИ-графа) представляет исходную задачу. Цель поиска на И/ИЛИ-графе — показать, что начальная вершина разрешима. Разрешимыми являются заключительные вершины (И-вершины), у которых разрешимы все дочерние вершины, и ИЛИ-вершины, у которых разрешимых вершин и указывает способ разрешимости начальной вершины. Наличие тупиковых вершин приводит к неразрешимым вершинам. Неразрешимыми являются тупиковые вершины, И-вершины, у которых неразрешима хотя бы одна дочерняя вершина, и ИЛИ-вершины, у которых неразрешима каждая дочерняя вершина.

Алгоритм Ченга и Слейгла. Основан на преобразовании произвольного И/ИЛИграфа в специальный ИЛИ-граф, каждая ИЛИ-ветвь которого имеет И-вершины только в конце. Преобразование использует представление произвольного И/ИЛИграфа как произвольной формулы логики высказываний с дальнейшим преобразованием этой произвольной формулы в дизъюнктивную нормальную форму. Подобное преобразование позволяет далее использовать алгоритм Харта, Нильсона и Рафаэля.

6.2.2 Метод ключевых состояний и ключевых операторов

Пространство описаний множеств подзадач представляется в виде специального направленного графа G, называемого И-ИЛИ графом или пропозиционным графом. Может ли такое представление стать основой для построения универсального решателя задач? Проведем небольшой анализ. Использование теоретикографической модели позволяет формализовать лишь один из элементов декларативного представления — пространства описания множества подзадач, а преобразования, определяемые на множестве вершин графа, заданы фиксированным множеством операторов, позволяющих на каждом шаге порождать все дочерние вершины для любой заданной. Таким образом, теоретико-графическая модель не предлагает подхода к решению в общем виде задачи или акта разбиения задачи на подзадачи. Эта модель позволяет решить задачу, если такой подход существует, т. е. если известны интерпретации задачи и ее подзадач, а также допустимые операторы разбиения задач на подзадачи. Так и были созданы на заре развития искусственного интеллекта системы (например, система интегрирования в символическом виде SAINT).

Механизм сведения задачи к подзадачам, независимым от задачи, назовем механизмом редукции. Одним из способов построения механизма редукции задачи в пропозиционном графе являются ключевые состояния и ключевые операторы [1, 9].

Н. Нильсон предложил метод сведения задачи к совокупности подзадач, последовательно упрощающий задачи поиска в пространстве состояний, т. е. накладывающий механизм редукции на решение задачи в системе продукций.

Представим задачу поиска в пространстве состояний в виде

$$(S_0, F, T)$$
,

где S_0 — множество начальных состояний,

Т — множество целевых состояний,

F — множество операторов, отображающих одно состояние в другое.

Пусть также заданы множества ключевых состояний T_1, T_2, \ldots, T_N , т. е. множества наиболее вероятных состояний решающего пути в графе. Тогда можно использовать механизм редукции для сведения задачи (S_0, F, T) к совокупности задач $(S_0, F, T_1), (\{t_1\}, F, T_2), \ldots, (\{t_N\}, F, T)$, где $t_1 \in T_1, t_2 \in T_2, \ldots, t_N \in T_N$ — конкретно выбранные ключевые состояния.

Одним из приемов нахождения множеств ключевых состояний является выделение ключевых операторов, т. е. операторов, применение которых необходимо для решения задачи. Пусть $f \in F$ — ключевой оператор для задачи (S_0, F, T) . Тогда задача может быть разбита на три подзадачи:

- 1) поиск пути к состоянию $t \in T_f, T_f$ область определения f, т. е. множество состояний, к которым f применим, подзадача (S_0, F, T_f) ;
- 2) применение оператора f подзадача $(\{t\}, F, \{f(t)\});$
- 3) оставшаяся часть задачи подзадача $(\{f(t)\}, F, T)$.

Недостатком описанного метода является то, что ключевые состояния или операторы могут быть найдены на основе анализа пространства состояний, а это очень сложная задача.

6.2.3 Метод анализа средств и целей

Относительно универсальным механизмом редукции задач является механизм, разработанный авторами GPS, который в той или иной модификации используется в современных планирующих системах. GPS (OP3) явился первой наиболее известной моделью планировщика. Он использовался для решения задач интегрального исчисления, логического вывода, грамматического разбора и др. OP3 объединяет два основных принципа поиска: анализ целей и средств и рекурсивное решение задач.

В GPS используется эвристический поиск. В процессе работы GPS находит различие между текущим и целевыми состояниями. На основе этих различий GPS выбирает оператор, который применяется к текущему состоянию, вырабатывая новое состояние. Далее производится сравнение этого состояния с целевым, и цикл повторяется. В случае неприменимости этого оператора к текущему состоянию, GPS определяет различия, суммирующие причину неприменимости. На основе этих различий выбирается оператор, пригодный для их устранения. Если он применим и устраняет их, то применяется предыдущий оператор. Схема работы GPS рекурсивна, т.к. оператор может быть неприменим или непригоден.

Механизм редукции задачи использует три стандартных метода [1] (рис. 6.1):

- преобразование состояния A в состояние B;
- уменьшение различия D между состояниями A и B;
- применение оператора f к состоянию A.

Преобразование состояния:

1) Генерируется выведенная (т.е. полученная путем последовательного

применения операторов к A и следующим состояниям) последовательность состояний, оканчивающаяся состоянием, идентичным B.

Уменьшение различия

2) Вырабатывается новое состояние A', выведенное из A с измененным разпичием D

Применение оператора

3) Гарантируется новое состояние применением f к A или состоянию, выведенному из A.

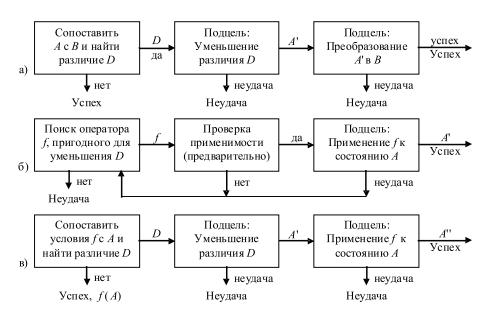


Рис. 6.1 – Основные методы механизма редукции задачи.

Представим пример работы механизма редукции на рис. 6.2, т. е. изобразим дерево методов для преобразования объекта A в объект B. Пытаясь преобразовать A в B, механизм находит различие D между A и B и переходит к его уменьшению (G_2) , находит оператор f_1 , пригодный для его уменьшения, и пытается применить его к $A(G_3)$. Однако оператор f_1 неприменим, и механизм находит различие D' и пытается его уменьшить (G_4) . Предположим, что оператор f_2 пригоден для уменьшения D' и применим к $A(G_5)$. Тогда вырабатывается новое состояние A'. Теперь механизм записывает A' как результат G_5 и G_4 и переходит к применению f_1 к A'. Поскольку различие D' устранено, f_1 применяется к A', вырабатывается результат A'' (G_6). Этот результат записывается в G_3 и G_2 . Поскольку различие D устранено, производится переход к преобразованию A'' в $B(G_7)$. К этому моменту механизм выработал последовательность операторов $f_2 \circ f_1(A) = f_1(f_2(A))$, преобразующую A в A'', и очередную подзадачу преобразования A'' в B.

Теперь представим процесс редукции этого примера с помощью пропозиционного графа (рис. 6.3).

В вершинах графа записаны формулировки исходной задачи и ее подзадач. Граф содержит только конъюнктивные вершины, так как мы предполагаем для простоты, что механизм GPS обладает способностью выбирать один пригодный оператор. Вершины S_3 и S_4 являются конечными, т. к. им соответствуют известные

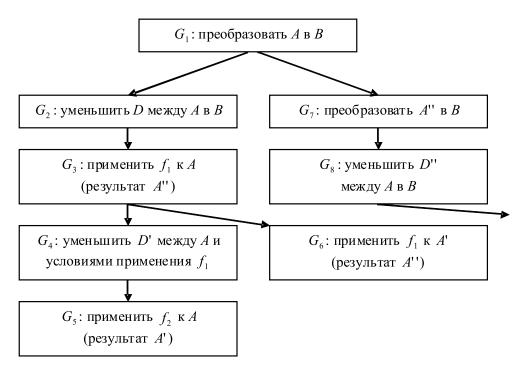


Рис. 6.2 – Пример работы механизма редукции.

операторы f_2 и f_1 соответственно, непосредственно преобразующие A в $A''(f_2)$ и A' в $A''(f_1)$.

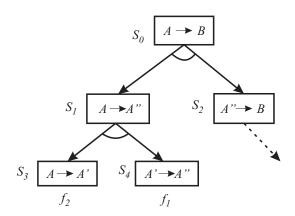


Рис. 6.3 – Пропозиционный граф, соответствующий процессу редукции.

Итак, мы сопоставили два формализма. Механизм редукции GPS осуществляет разбиение задачи на подзадачи с помощью метода уменьшения различия, но должен обладать методами определения различия и выбора оператора для уменьшения или устранения различия. Однако этот механизм не всегда гарантирует нахождение решения. Можно лишь показать, что наличие упорядоченного решения задачи эвристического поиска, с треугольной таблицей связок (группы операторов, являющихся пригодными для уменьшения различий) и всех ее подзадач дает достаточные условия того, что механизм редукции GPS найдет решения задачи, если оно есть.

Пропозиционный граф, как мы уже говорили, использует теоретико-графическую модель и формализует пространства множеств подзадач, но без информации

о конкретном содержании задачи и о свойствах пространства описания множества подзадач мы не можем определить множество операторов, преобразующих вершины графа в дочерние вершины. Можно показать, что представление задачи в виде пропозционного графа допускает построение допустимых алгоритмов, т. е. всегда находящих решение задачи, если оно есть, а иногда даже оптимальных решений.

6.2.4 Планирование с помощью логического вывода

Такое планирование предполагает: описание состояний в виде правильно построенных формул (ППФ) некоторого логического исчисления, описание операторов в виде либо ППФ, либо правил перевода одних ППФ в другие. Представление операторов в виде ППФ позволяет создавать дедуктивные методы планирования, представление операторов в виде правил перевода — методы планирования с элементами дедуктивного вывода.

Дедуктивный метод планирования системы QA3. ОР3 не оправдал возлагавшихся на него надежд в основном из-за неудовлетворительного представления задач. Попытка исправить положение привела к созданию вопросно-ответной системы QA3. Система рассчитана на произвольную предметную область и способна путем логического вывода ответить на вопрос: возможно ли достижение состояния В из A? В качестве метода автоматического вывода используется принцип резолюций. Для направления логического вывода QA3 применяет различные стратегии, в основном синтаксического характера, учитывающие особенности формализма принципа резолюций. Эксплуатация QA3 показала, что вывод в такой системе получается медленным, детальным, что несвойственно рассуждениям человека.

Метод продукций системы STRIPS. В этом методе оператор представляет продукцию $P, A \Rightarrow B$, где P, A и B- множества ППФ исчисления предикатов первого порядка, P выражает условия применения ядра продукции $A \Rightarrow B$, где B содержит список добавляемых ППФ и список исключаемых ППФ, т. е. постусловия. Метод повторяет метод OP3 с тем отличием, что стандартные задачи определения различий и применения подходящих операторов решаются на основе принципа резолюций. Подходящий оператор выбирается так же, как в OP3, на основе принципа «анализ средств и целей». Наличие комбинированного метода планирования позволило ограничить процесс логического вывода описанием состояния мира, а процесс порождения новых таких описаний оставить за эвристикой «от цели к средству ее достижения».

Метод продукций, использующий макрооператоры. Макрооператоры — это обобщенные решения задач, получаемые методом STRIPS. Применение макрооператоров позволяет сократить поиск решения, однако при этом возникает проблема упрощения применяемого макрооператора, суть которой заключается в выделении по заданному различию его требуемой части и исключении из последней ненужных операторов.

6.3 Примеры планирующих систем

Планирующая система STRIPS. STRIPS—Stanford Research Institute Problem Solver (решатель задач Стэнфордского исследовательского института).

PLANEX — Plan Executive (исполнитель планов).

Кратко опишем планирующую систему для робота Стэнфордского исследовательского института. Это широкий класс универсальных решателей задач для интеллектуальных роботов (ИР), планирующая система (ПС) которых представляет проблему планирования и выполнения действий и возможных путей ее решения (рис. 6.4) [1].

Робот Стэнфордского исследовательского института действует в мире, состоящем из комнат с дверьми и предметами (ящики, призмы), и способен осуществлять с этими предметами относительно простые манипуляции.

Мир робота представляется в ΠC в виде модели, состоящей из набора правильно построенных формул ($\Pi \Pi \Phi$) в исчислении предикатов первого порядка, описывающих состояние мира в данный момент.

Действия робота моделируются множеством операторов, определяемых наименованием, списком параметров, условиями применимости (предусловиями) и результатами действия в виде схем $\Pi\Pi\Phi$, т. е. $\Pi\Pi\Phi$, зависящих от параметров.

Результаты действия операторов описываются списком вычеркивания тех схем $\Pi\Pi\Phi$, которые перестают быть истинными после применения оператора, и списком добавлений схем $\Pi\Pi\Phi$, которые становятся истинными после применения оператора.

Пусть имеется некоторая целевая схема ППФ G(P), где P—множество параметров схемы, которую нужно доказать на множестве M дизьюнктов, т. е. найти опровержение множества дизьюнктов $M \cup \{ {}^{\sim} G(P) \}$. Для вычисления частного случая P' множества P, при котором множество $M \cup \{ {}^{\sim} G(P) \}$ невыполнимо, можно использовать стандартный алгоритм унификации. С помощью этого алгоритма можно найти наиболее общие частные случаи параметров, при которых обеспечивается унификация. Однако необходимо определить, какие подстановки допустимы в случае параметров. Определим следующие типы термов, которые могут быть подставлены вместо переменной: переменные, константы, параметры и функциональные термы, не содержащие переменных. Вместо параметров могут быть подставлены следующие типы термов: константы, параметры и функциональные термы, не содержащие функций Сколема, переменных или параметров.

Поскольку один и тот же параметр может иметь несколько вхождений в множестве дизьюнктов, он должен замещаться при резолюции термом во всех дизьюнктах, являющихся производными от резольвенты.

Метод доказательства теорем используется только внутри модели мира для ответов на вопросы, связанные с анализом применимости операторов и проверкой выполнимости условий достижения цели.

Для поиска решений в пространстве моделей ПС STRIPS используется механизм редукции GPS. Почему? В практических системах на применение любых формализмов описания накладываются, как обычно, ограничения количественного порядка. Предположим, что в поисках плана решения задачи робот имеет в своем распоряжении в среднем 6 операторов, применимых и эвристически обоснованных в каждом состоянии; предположим, что типичная задача решается последовательностью из 4-х операторов, тогда поисковое дерево будет иметь около $6^4 \approx 1300$ вершин. Будем считать, что робот работает в среде умеренной сложности. Тогда для полного описания каждого состояния может потребоваться хранение около 1000 элементарных фактов, касающихся местонахождения всех предметов,

указания всех отношений между ними и т. д. Оказывается, что только для описания всех состояний в поисковом дереве требуется хранить свыше миллиона фактов. Но каждый факт сам представляет сложную структуру. Каждое действие, совершаемое оператором, вызывает изменения и т. д.

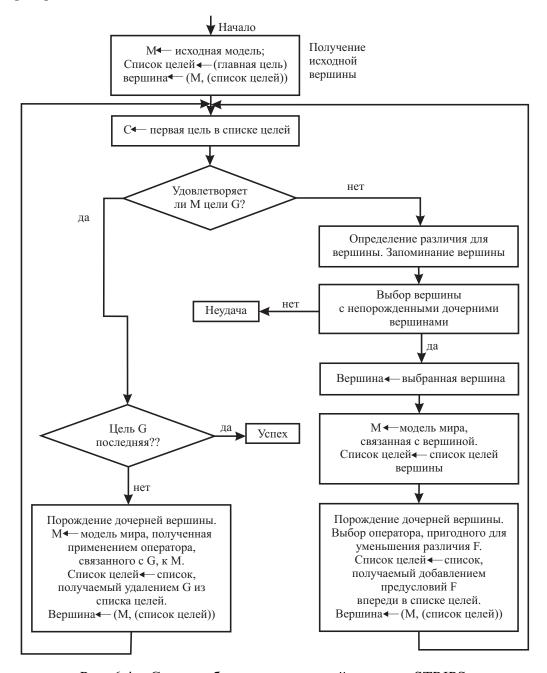


Рис. 6.4 – Схема работы планирующей системы STRIPS.

Маккарти и Хэйс приводят пример проблемы полных описаний. Для того чтобы человек p вступил в телефонный разговор с человеком q, казалось бы, достаточно чтобы p нашел номер телефона q в телефонной книжке и набрал его. Решатель задач потерпит неудачу, если:

- страница с номером телефона q будет вырвана;
- человек р слепой;

- будет ошибка в коммутации;
- человек р заболел и потерял голос;
- телефон p не значится в телефонной книге и т. д.

В процессе поиска механизм редукции порождает иерархию целей, подцелей и моделей, которую можно представить в виде дерева поиска. Каждая вершина дерева имеет вид (модель, (список целей)) и соответствует задаче достижений по порядку подцелей из списка целей указанной модели среды. Схема работы ПС представлена на рисунке 6.4. В ПС STRIPS имеется эвристический механизм выбора узлов дерева поиска. Это оценочная функция, учитывающая такие факторы, как число оставшихся целей в списке целей, число и тип предикатов в оставшихся выражениях цели, а также сложность различий, связанных с данным узлом.

Главной трудностью использования механизма редукции GPS является выбор операторов, пригодных для устранения или уменьшения различий. В нашей системе эта трудность решается следующим образом. Предположим, что в дереве поиска образован узел $(M, (G_i, G_{i-1}, \ldots, G_0))$, причем система доказательства теорем пытается доказать невыполнимость множества $M \cup \{\sim G_i\}$. Если доказательство успешно, то к модели M применяется оператор с предусловиями G_i . Если же в течение определенного времени опровержение не будет найдено, то незавершенное доказательство или, в случае, если оно велико, его часть, выбираемая из эвристических соображений, и берется в качестве различия между M и G_i и связывается с данным узлом.

Процесс выбора пригодного оператора происходит в два шага. На первом шаге создается упорядоченный список операторов кандидатов. Выбор кандидатов основан на простом сравнении предикатов в различии с предикатами списков добавления операторов. На втором шаге программа доказательства теорем определяет, могут ли высказывания из списка добавлений оператора резольвировать с высказываниями в различиях. Если новые резольвенты являются производными от высказываний в списке добавлений, то соответствующий оператор объявляется пригодным. Заметим, что из одного оператора-схемы может быть образовано несколько пригодных частных случаев.

Обобщение планов и планирования с помощью макрооператоров. Задача обобщения планов состоит в том, чтобы после построения успешного плана преобразовать этот конкретный план в план, который мог бы быть использован для множества подобных задач. Другими словами, мы хотим получить план-схему, т. е. параметризованное семейство планов.

Реализация алгоритмов планирования. Одной из важных компонент в современных интеллектуальных системах, в частности в системах поддержки принятия решений, является подсистема планирования. В настоящее время разработаны различные методы планирования, основанные на исчислении предикатов и реализованные с использованием языка Пролог. Рассмотрим реализацию алгоритмов планирования действий робота на языке Пролог, на которой продемонстрируем основные моменты решения задач данного класса [2].

Для решения задач планирования обычно выделяются 2 типа состояний: начальное состояние и конечное состояние, описываемые предикатами. Для нашего примера начальным состоянием является следующее расположение кубиков и манипулятора (hand): кубики «В» и «С» находятся на столе, кубик «А» стоит на

кубике «В», манипулятор расположен над кубиком «С». Конечным состоянием является следующее расположение кубиков: кубики «А» и «В» находятся на столе, а кубик «С» стоит на кубике «В». Начальное и конечное состояния описываются следующими списками предикатов:

Далее строится пространство состояний, начальные уровни которого представлены на рис. 6.5. Затем задается предикат, описывающий перестановку кубиков. Этот предикат имеет имя «move», и он задает 4 возможности для перестановки кубиков, описанные с использованием предикатов «add» и «del», которые используются для добавления и удаления из списка заданных элементов.

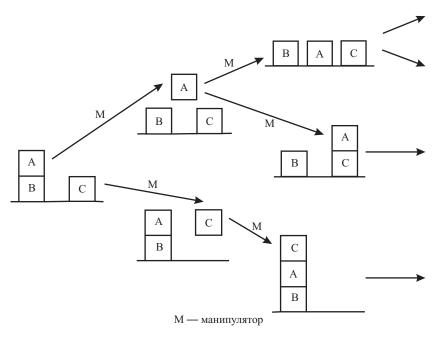


Рис. 6.5

Предикат move имеет три аргумента. Первый из них обозначает имя действия, которое выполняется со своими аргументами. Второй аргумент представляет список предусловий: предикаты, которые должны быть истинны в описании состояния мира, к которому применяется правило перестановки кубиков, заданное предикатом move. Третьим аргументом является список добавляемых и удаляемых предикатов, т. е. предикатов, которые добавляются к состоянию мира или удаляются из него для того, чтобы создать новое состояние как результат применения данного правила.

Опишем действия, выполняемые рекурсивным генератором плана или планировщиком.

- 1) Поиск предиката move.
- 2) Проверка для предиката move предусловий.
- 3) Предикат change_state создает новое состояние (Child_state), используя списки предикатов для добавления и удаления.
- 4) Предикат member_stack проверяет, не встречалось ли полученное состояние ранее.
- 5) Оператор stack заносит новое состояние (Child_state) в New_moves_stack.
- 6) Оператор stack заносит новое имя состояния (name) в New been stack.
- 7) Рекурсивно вызывается предикат plan для поиска следующего состояния, при этом используется Child_state и обновленные New_moves_stack и New_been_stack.

Процедурой поиска является метод поиска в глубину с возвратом, реализованный в языке Пролог и заканчивающийся при нахождении целевого состояния.

Приведем программу реализации планировщика.

```
plan(State, Goal, _, Move_stack) :- equal_set(State, Goal),
    write(>>moves are>>), nl, reverse_print_stack(Move.Stack).
plan(State, Goal, Been_stack, Move_stack) :- move(Name,
    Preconditions, Actions), conditions_met(Preconditions,
    State), change_state(State, Actions, Child_state),
    not(member_stack(Child_state, Been_stack)), stack(Name,
    Been_stack, New_been_stack), stack(Cbild_state, Move_stack,
    New_move_stack), plan(Child_state, Goal, New_been_stack,
    New_move_stack),!.
plan(_, _, _) :- write(>>No plan possible with these moves!>>).
conditions_met(P, S) :- subset(P, S).
change_state(S, [], S).
change_state(S, [add(P) | T], S_new :- change_state(S, T, S2),
             add_if_not_jn_set(P, S2, s_new),!.
change_state(S, [del(P) | T] , S_new) :- change_state(S, T, S2),
             delete_if _in_set(P, S2, S_new),!.
reverse_print_stack(S) : - empty_stack(S).
reverse_print_stack(S) : - stack(E, Rest, S)
reverse_print_stack(Rest), write(E), nl.
```

Зададим предикат до для инициализация аргументов предиката plan, также

опишем предикат test для задания тестов, содержащих начальное и целевое состояния.

```
go(Start, Goal) :- empty_stack(Move_stack),
empty_stack(Been_stack),
    stack(Start, Been_stack, New_been_stack), plan(Start, Goal,
    New_been_stack, Move_stack).

test :- go([hand_empty, on_table(b), on_table(c), on(a, b),
    clear(c), clear(a)],
    [hand_empty, on_table(a), on_table(b), on(c, b), clear(a),
    clear(c)]).
```

Результатом работы программы планировщика является список действий на экране монитора, которые необходимо выполнить для достижения заданного целевого состояния из заданного начального состояния.

6.4 История интеллектуального планирования

Интеллектуальное планирование зародилось на основе исследований в области поиска в пространстве состояний и автоматического доказательства теорем. STRIPS была первой системой, которая базировалась на этих двух методологиях. Система STRIPS создавалась в качестве модуля планирования для мобильного робота Shakey (проект Стенфордского исследовательского центра SRI). Поиск решения в этой системе был основан на методологии поиска системы GPS, а для проверки истинности предусловий действия в текущем состоянии применялась методология доказательства теорем из системы QA3.

В 1957 году была начата разработка системы, получившей название «Универсальный решатель задач» (General Problem Solver — GPS). Авторы этой системы: Алан Ньюэлл, Дж. Шоу и Герберт Саймон.

GPS (Newell, Shaw, Simon, 1957)

Название «Универсальный решатель задач» обусловлено тем фактом, что это была первая система для решения задач, в которой общие методы решения задач были отделены от знаний, определяющих конкретную задачу. Часть программы, осуществлявшая поиск решения, не обладала информацией о конкретном виде задачи, с которой она работала. Специфичные для конкретной задачи знания организовывались в отдельные структуры: объекты и операторы для преобразования объектов. Задача для системы GPS ставилась в виде пары, состоящей из начального объекта и желаемого объекта, в который начальный объект должен быть преобразован.

Методология, использованная в GPS, отличается от стандартных методов поиска в пространстве состояний тем, что она выбирает путь, по которому продолжить поиск. То есть система пытается искать решение в первую очередь в «наиболее перспективных» ветвях поиска. Такая методология была названа «анализ средств и целей». Ее суть заключалась в том, что сначала отыскивалось различие между

текущим объектом и объектом, который мы хотим получить. Это различие относилось к одному из ряда классов различий. С каждым классом был сопоставлен набор действий, способных уменьшить различие между текущим и целевым объектами.

Несмотря на то, что в качестве объекта могла выступать модель мира, а в качестве оператора — действие, задача планирования перед системой GPS не ставилась. Такой вариант использования GPS был осуществлен значительно позднее, когда была разработана система STRIPS.

QA3 (Green, 1969)

Система QA3 хотя и была вопросно-ответной системой, однако содержала в себе еще и механизмы для решения задач. В частности, эта система могла отвечать на вопросы такого типа: «Существует ли последовательность действий, такая, что объект object 1 окажется в комнате room4, если изначально он находится в комнате room2?» Такая формулировка вопроса является, по сути, постановкой задачи планирования. Но так как целью системы QA3 были все-таки ответы на вопросы, а не решение задач, да и реализация механизма решения задач имела множество ограничений, эта система не претендует на звание первого планировщика.

STRIPS (Fikes and Nilsson, 1971)

Первая система, которая создавалась именно для решения задачи планирования.

Файкс и Нильсон сформулировали задачу планирования такой, какова она и по сей день. Задача планировщика найти такую последовательность действий, которая преобразует начальное состояние в такое состояние, в котором достигается заранее заданное целевое условие. Состояние представляется множеством формул логики первого порядка. Цель также описывается формулой. Действия описываются в виде специальных конструкций. В STRIPS поиск в пространстве состояний осуществляется по аналогии с системой GPS, а именно: используется стратегия «анализ средств и целей». Для осуществления поиска плана была использована стратегия GPS, которая определяет «различие» между текущим состоянием и целью и находит действия, способные уменьшить это различие. После того, как подходящее действие найдено, мы решаем подзадачу порождения состояния, к которому данное действие применимо. Если такое состояние обнаруживается, то действие применяется и изначальная цель рассматривается вновь уже в новом состоянии. Когда действие признано подходящим (релевантным), мы еще не знаем, где оно появится в конечном плане. То есть может оказаться, что перед ним должны стоять другие действия, которые делают возможным выполнение данного действия. После него также могут оказаться действия, которые обеспечивают выполнение всей цели (ведь одно действие служит уменьшению различия, а не обязательно достижению цели сразу).

Формально, постановка задачи в STRIPS включает три составляющие: начальное состояние, множество действий и целевую формулу.

Впоследствии STRIPS подвергался обширной критике и за подход к описанию мира и действий, и за алгоритм поиска.

STRIPS+MACROP (Fikes, Hart, Nilsson, 1972)

Уже в следующем году Файкс и Нильсон совместно с Питером Хартом представили доработку своей системы. В своей статье они описывают механизм для обобщения планов, порожденных STRIPS. Обобщенные планы хранились в специальных треугольных таблицах и использовались как своеобразные «макрооператоры» для повторного применения в других задачах планирования, а также для мониторинга исполнения плана. Получение и сохранение обобщенных планов являлось некоторого рода формой обучения, которое позволяло уменьшить время, необходимое на поиск плана в схожих задачах.

В своем первозданном виде STRIPS выдает в качестве результата полностью конкретизированный план. Ключевым моментом доработки к STRIPS стало обобщение плана путем замены констант параметрами. Другими словами, конкретный план становится схемой плана или макрооператором, который подобен примитивным схемам действий, заданным в задаче изначально.

HACKER (Sussman, 1973)

Система HACKER основана на методе анализа средств и целей и обладает механизмом обучения.

В качестве механизма обучения Зюссман использовал упрощенный вариант обучения по прецедентам. Прецеденты сохранялись в библиотеке прецедентов, названной Зюссманом библиотекой ответов. Когда система HACKER решала очередную задачу, она сначала проверяла, нет ли в библиотеке ответов решения, чей образец применимости соответствует условиям задачи. Если такое решение находилось, то оно применялось. Если нет, то HACKER находил новое решение. При построении нового решения система могла использовать библиотеку ответов для разрешения частных подпроблем, а также старалась избегать ошибок, с которыми она сталкивалась ранее. Новое решение сохранялось в библиотеке ответов, проиндексированное по образцу применимости, выведенному из постановки задачи. Таким образом, оно могло быть использовано для решения подобных задач в будущем. Если решение в момент применения приводило к ошибке, то использовались общие механизмы отладки для классификации и установления природы ошибки. После этого решение редактировалось и перезаписывалось в библиотеке ответов. Часто ошибка сама подвергалась обобщению и запоминанию, для того, чтобы избегать ее уже при построении решения.

НАСКЕR—это попытка совместить в одной системе процедурное и декларативное представления знаний о предметной области, чтобы получить одновременно легко расширяемый (декларативное представление) и эффективный (процедурное представление) решатель задач. Ряд проблем система «знает», как решать (эти знания могут приобретаться, например, посредством обучения). Метод решения—это процедурное представление знаний. Процедура описывает порядок шагов и способ избегания негативного взаимовлияния шагов. Процедурное представление обеспечивает высокую скорость поиска известных решений. Декларативное представление знаний служит залогом хорошей расширяемости описания предметной области. В этой форме могут быть легко добавлены новые факты о мире (объекты, отношения).

WARPLAN (Warren, 1973)

Еще одно решение проблемы с чередованием шагов в нелинейных планах было предложено Дэвидом Уорреном. Его подход заключался в отказе от STRIPS-овского варианта анализа средств и целей. Вместо него был предложен несколько иной подход, названный регрессией. В STRIPS поиск подходящих действий велся от цели, но построение плана выполнялось от начального состояния. Регрессия осуществляет построение плана с конца (от цели к начальному состоянию). При этом рассматриваются все возможные способы достижения каждой из подцелей и все возможные порядки их достижения. Достигнутые цели не должны были разрушаться в текущей ветви поиска. Именно рассмотрение всех возможных порядков позволяет преодолеть проблему с чередованием.

Планировщик WARPLAN, реализующий этот подход, был написан Уорреном на Прологе и состоял всего лишь из ста строк кода.

INTERPLAN (Tate, 1974)

Остин Тейт предложил подход, основанный на переупорядочивании подцелей в процессе планирования. Реализация этого подхода воплотилась в планировщике INTERPLAN.

В своей работе Тейт рассматривает в качестве цели конъюнкцию высказываний, выраженных атомарными формулами (будем называть эти высказывания целями-конъюнктами) [5]. Цели-конъюнкты достигаются в определенной последовательности и, будучи достигнутыми, уже не должны разрушаться до конца процесса планирования. Тейт вводит понятие «период сохранности» цели-конъюнкта, которым обозначает интервал времени от момента достижения цели-конъюнкта (когда она стала окончательно истинной) до момента достижения целевого состояния.

Планировщик INTERPLAN (interleaving planning) учитывает необходимость (в ряде случаев) перемещения подцелей по временной оси. Тейт пишет, что на самом деле нужно рассмотреть лишь небольшое число возможных чередований. Следует пользоваться допущением линейности, т.к. это очень мощная эвристика, но в случае обнаружения противоречий следует учитывать возможность чередования. Если противоречие обнаружено, то можно либо переставить цели более высокого уровня, либо попробовать достичь их в противоположном порядке (например, поместить вторую цель непосредственно перед первой).



Итак, INTERPLAN—планировщик на основе анализа средств и целей с дополнительной эвристической функциональностью, позволяющей более гибко упорядочивать цели-конъюнкты и подцели, чем это делал STRIPS. Этот планировщик также не обладает полнотой и опирается на допущение о линейности. Однако более развитые эвристические средства по манипулированию порядком целей позволили ему справиться с аномалией Зюссмана.

ABSTRIPS (Sacerdoti, 1974)

Эрл Сасердоти предложил использовать иерархию абстрактных пространств поиска для того, чтобы на каждом уровне абстракции отделять действительно важную информацию от несущественных в данный момент деталей и тем самым отсекать нерелевантные ветви поиска уже на верхних уровнях абстракции. Его идеи легли в основу планировщика ABSTRIPS (Abstraction-Based STRIPS)—STRIPS-подобного планировщика с поддержкой идеи абстрактных пространств поиска.

Абстрактное пространство поиска должно соответствовать двум критериям. С одной стороны, оно должно в достаточной степени отличаться от детального пространства для значительного повышения производительности. С другой стороны, оно не должно быть настолько абстрактным, чтобы затруднить отображение абстрактного пространства в конкретное (детальное).

Кроме этого, в ABSTRIPS реализована отложенная инстанциация параметров действий и эвристика выбора следующего узла в дереве поиска, учитывающая не только расстояние до цели, но и расстояние от начального состояния (во избежание построения излишне длинных планов).

NOAH (Sacerdoti, 1975)

NOAH — первый планировщик, допускающий частичное упорядочение действий в плане. При таком подходе, план — это набор отношений порядка между действиями. Связный план можно получить из набора частично упорядоченных действий путем линеаризации. Кроме этого, такой подход делает возможным наличие параллельных ветвей в плане.

NOAH — первый иерархический планировщик. Иерархическое планирование предполагает, что процесс составления плана — это процесс постепенной детализации решения. Действия в NOAH могут быть простыми и составными (раскладываться на набор действий-составляющих). Составные действия содержат в себе знания о типовых решениях часто встречающихся задач.

Планировщик использовался в «Системе компьютерной консультации» (Computer Based Consultant Project).

NONLIN (Tate, 1977)

Очередной важной вехой в истории планирования стал планировщик NONLIN (Non-linear planner). Идеологически этот планировщик является развитием планировщика NOAH. Однако Тейт вводит новый термин—задача, — положивший начало новому направлению, называемому HTN-планированием (HTN—hierarchical task network—иерархическая сеть задач).

Планировщик NONLIN появился как часть проекта «The Planning: a joint AI/OR approach». Целью этого проекта было создание программы, помогающей пользователям составлять сетевые графики проектов (project network).

NONLIN обходит пространство поиска полностью (в отличие от NOAH). Если в какой-то момент времени была выбрана неверная детализация (метод решения), то будет выполнен откат и выбор альтернативных вариантов.

В NONLIN используются идеи INTERPLAN для разрешения проблемы чередования.

SIPE (Wilkins, 1983)

Этот планировщик известен тем, что осуществляет планирование ресурсов, т. е. работает не только с логическими выражениями, но и с численными величинами. Относится к числу HTN-планировщиков. Разработан Дэвидом Уилкинсом (David Wilkins).

TWEAK (Chapman, 1987)

TWEAK — еще один планировщик, использующий частичное упорядочивание действий в плане. Однако, в отличие от NOAH и NONLIN, он не является иерархическим.

Чепмен рассматривал планирование не как процесс последовательной генерации шагов, а как процесс наложения ограничений, которым должен соответствовать результирующий план. ТWEAK он называет планировщиком, накладывающим ограничения. Наложение ограничений—это процесс определения объекта, в данном случае плана, путем постепенного перечисления частичных ограничений, которым он (объект) должен соответствовать. С другой стороны, наложение ограничений можно рассматривать как стратегию поиска, при которой, вместо генерирования и проверки отдельных ветвей поиска, целые куски пространства поиска постепенно удаляются из рассмотрения (посредством ограничений) до тех пор, пока не останутся только альтернативы, удовлетворяющие условиям поиска.

Такой подход можно рассматривать как постепенное уточнение незавершенного плана. Изначально незавершенный план состоит из описания начального состояния и не содержит действий и ограничений. В процессе планирования план уточняется путем добавления схем действий и ограничений (на порядок следования схем действий и значения их параметров). Незавершенный план может быть завершен несколькими разными способами (в зависимости от того, какие схемы действий будут добавлены и какие ограничения будут наложены в дальнейшем), т. е. он соответствует классу завершенных планов.

Архитектурно TWEAK можно разделить на две составляющие: управляющую процедуру и процедуру достижения цели. Первая из них в цикле осуществляет выбор очередной подцели. Изначальная цель может состоять из нескольких подцелей. Кроме этого, добавление в план новых действий может приводить к появлению новых подцелей, обусловленных предусловиями этих действий. Перебор целей осуществляется в ширину, т. е. сначала рассматриваются способы достижения исходных подцелей, потом подцелей, появившихся в результате добавления действий для достижения исходных подцелей, и т. д. Если очередная цель оказывается недостижимой, то выполняется откат и ищется альтернативное решение. Таким образом, осуществляется исчерпывающий поиск, т. е. если план существует, то он будет найден.

Вторая процедура — процедура достижения цели — обеспечивает достижимость одной определенной цели, если это возможно. Эта процедура опирается на критерий модальной истинности, определяющий условия, когда некоторое

высказывание гарантированно будет иметь значение «истина». Процедура достижения цели обеспечивает выполнимость этого условия для достигаемой цели.

ABTWEAK (Yang, Tenenberg, 1990)

POP-планировщик (Partial Order Planning), т. е. планировщик с частичным упорядочиванием действий.

Авторы перенесли идею иерархии абстрактных пространств ABSTRIPS в планировщик с частичным упорядочением действий (а именно, TWEAK).

PABLO (Christensen, 1990)

PABLO — планировщик, использующий абстрагирование (путем ослабления цели и предусловий действий).

CHEF (Hammond, 1990)

СНЕF — планировщик, опирающийся на методологию рассуждений по прецедентам. Прикладная область — кулинария — приготовление блюд, обладающих определенными свойствами.

В качестве прецедента сохраняется не только информация об успешных эпизодах планирования, но и информация о сбоях, возникших в процессе получения плана. Последнее позволяет предсказывать и избегать возникновения подобных ситуаций в будущем. При хранении планы индексируются по целям, достижению которых они служат, и по проблемам, которых следует избегать, если пользоваться данным решением. Система учится сразу трем аспектам:

- новым планам,
- свойствам, по которым можно предсказать возникновение проблем,
- методам ремонта, который должен быть выполнен в случае возникновения таких проблем в новых обстоятельствах.

Как и большинство других CBR-систем, CHEF строит новые планы на основе хранящихся в памяти прецедентов. Прежде чем искать подходящий план и модифицировать его, CHEF исследует цели и предсказывает сбои, которые могут возникнуть из-за взаимовлияния планов, которые будут использованы для достижения этих целей. Это предсказание выполняется на основе проблем, которые планировщик встречал в прошлом. Когда сбой предсказан, CHEF добавляет цель «избежать этого сбоя» в изначальный список целей. CHEF умеет не только предсказывать сбои и избегать их, но и ремонтировать планы в случае, если сбой все-таки возникнет.

SNLP(Soderland, Weld, 1991)

Планировщик SNLP примечателен своей простотой как в плане формализации, так и в отношении самого алгоритма. Основан на методологии формирования частичных планов.

O-PLAN (Currie, Tate, 1991)

В системе O-PLAN реализован иерархический планировщик, имеющий временное представление и способный манипулировать ограниченными ресурсами.

UCPOP (Penterthy, Weld, 1992)

Полный и корректный РОР-планировщик, поддерживающий подмножества ADL, а именно условные эффекты, кванторы всеобщности в предусловиях, эффектах и целях. Планировщик полноценно реализует слабое связывание: частичный порядок действий и отложенное означивание переменных.

Одна из наиболее развитых систем планирования, использующая нелинейный и иерархический подходы и обладающая распределенностью, — планирование производится для нескольких агентов.

SATPLAN (Kautz, Selman, 1992)

В 1992 году Генри Каутцем и Бартом Селманом был предложен принципиально иной подход к задаче планирования. Идея заключалась в следующем. Сначала задача планирования формулируется в виде множества ограничений, выраженных логическими формулами. Ограничения накладываются таким образом, чтобы всякая модель этого множества формул соответствовала корректному плану. Затем находится модель для получившегося множества формул, и уже из модели извлекается готовый план.

Планировщик всегда находит план минимальной длины.

Основным недостатком такого подхода является огромное количество и большие размеры формул, получающихся в результате кодирования задачи планирования в SAT-задачу.

Алгоритм способен справляться с большими задачами в сложных (переборных) предметных областях.

BURIDAN (Hanks, Weld, 1993)

В 1990-95 годах происходит всплеск развития вероятностных планировщиков. В отличие от классических планировщиков, предполагающих детерминированность поведения и точные знания о мире, вероятностные планировщики адаптированы для рассуждений в средах с неопределенностью. Одной из наиболее известных работ в этом направлении является планировщик BURIDAN.

В BURIDAN источниками неопределенности являются неточные знания о начальном состоянии и эффектах действий. Вместо классического начального состояния в постановке задачи планирования фигурирует распределение вероятностей на множестве возможных начальных состояний. Кроме того, эффекты действий зависят от случайных факторов (вероятности проявления того или иного эффекта задаются в описании схем действий). Использование вероятностной модели усложняет также определение корректного плана. Вместо классического «плана, достигающего цель», используется «план, достигающий цель с вероятностью

выше заданного порога». Алгоритм BURIDAN основан на алгоритме SNLP и является полным и корректным.

Начиная с пустого плана, BURIDAN осуществляет две операции:

- 1) Оценку стоимости плана. На этом этапе проверяется, превышает ли вероятность того, что текущий план достигает цель, заданный порог Т. Если это так, то работа алгоритма завершается и возвращается план. Этот шаг сложен в вычислительном аспекте. Было разработано несколько стратегий для оценки, каждая из которых выгодна в одних случаях и неэффективна в других.
- 2) Уточнение плана. Выполняется попытка увеличить вероятность достижения цели путем уточнения (ликвидации угрозы или добавления причинной связи, увеличивающей вероятность). В случае добавления причинной связи все аналогично SNLP. Однако для ликвидации угрозы, кроме обычных повышения и понижения, используется дополнительный способ разрешения угроз конфронтация. Суть конфронтации заключается в том, чтобы сделать невозможным выбор заключений, представляющих угрозу. Точнее выбираются только те заключения, которые не представляют угрозы, а триггеры этих заключений становятся подцелями. Подобный прием использовался и в UCPOP для условных эффектов.

DerSNLP (Ihrig, Kambhampati, 1994)

DerSNLP — DERivational SNLP — планировщик, опирающийся на методологию рассуждений по прецедентам.

В качестве прецедентов хранятся не конечные планы, а способы их получения (трассы вывода решений). Предварительное решение получается при помощи механизма «переигрывания». Затем это решение адаптируется при помощи алгоритма SNLP. То есть переигрывание дает нам новое начальное состояние (предположительно находящееся ближе к решению) и формирует некоторый фрагмент плана, а затем работает обычный планировщик. Трасса вывода всегда переигрывается до конца (хотя некоторые инструкции трассы могут пропускаться, в силу того что не выполнимы в новых условиях).

GRAPHPLAN (1995)

GRAPHPLAN имеет два основных этапа, каждый из которых чередуется в цикле. Прежде всего этот планировщик проверяет, присутствуют ли все целевые литералы на текущем уровне без взаимно исключающих связей между любой парой из них. Если это требование соблюдается, то в текущем графе может существовать решение, поэтому планировщик выполняет попытку извлечь это решение. В противном случае граф расширяется путем добавления действий для текущего уровня и литералов состояния для следующего уровня. Процесс продолжается до тех пор, пока либо не обнаруживается решение, либо не выясняется, что решения не существует.

IPP (Koehler, 1997)

IPP — развитие планировщика GRAPHPLAN. IPP поддерживает условные эффекты и кванторы всеобщности в эффектах, а также более сложные (по сравнению с GRAPHPLAN) предусловия. Кроме этого, реализованы методы RJFO hGAM. С 1998 года поддерживает метрическое планирование и планирование с ресурсами. Планировщик реализован на языке Си.

HSP (Bonet, Geffner, 1998)

HSP (Heuristic Search Planner) — планировщик, выполняющий прямой поиск в пространстве состояний (то есть от начального состояния к конечному) и использующий независимую от домена эвристику для оценки расстояния до цели, причем знания, на которые опирается эвристика, автоматически извлекаются из описания задачи планирования.

Планировщик работал в рамках STRIP S-доменов, предварительно выполнялась инстанциация всех схем действий. Для расчета эвристической функции выполнялись следующие шаги. Сначала задача упрощалась путем отбрасывания всех списков удалений в действиях. Таким образом, добавление очередного действия в последовательность действий могло приводить лишь к увеличению числа литералов в описании состояния. Затем, начиная от начального состояния, строились все возможные цепочки действий. При этом для каждого литерала в состоянии рассчитывалась оценка минимального количества шагов в плане, необходимых для достижения этого литерала. Алгоритм построения цепочек действий останавливался, когда ни одна из оценок не изменялась при добавлении очередного действия. Эвристическая функция, определяющая стоимость каждого из возможных в данном состоянии шагов, рассчитывалась как сумма таких оценок для каждой из целей задачи планирования. Поиск плана осуществляется каким-либо информированным алгоритмом, использующим эту эвристическую функцию.

HSPr — регрессионная версия планировщика HSP; то есть использовался не прямой, а обратный поиск в пространстве состояний.

ALTALT (Kambhampati, Nguyen, Nigenda, 2000)

AltAlt (абревиатура от «A Little of This a Little of That») — универсальный планировщик, представляющий собой объединение методологии планировщика GRAPHPLAN и эвристического планирования в пространстве состояний. AltAlt использует структуру графа планирования (из GRAPHPLAN) для построения эффективных эвристик, которые затем используются планировщиком в пространстве состояний для выбора перспективных ветвей поиска.

FF (Hoffmann, Nebel, 2000)

FF (сокр. от FastForward) — планировщик, являющий собой развитие идей планировщика HSP (прямой поиск и эвристическая оценка расстояния до цели без учета списков удалений в действиях). В отличие от HSP функция расчета эвристических оценок этого планировщика принимает во внимание то, что однажды

достигнутый факт может являться предусловием нескольких последующих действий. Для выполнения этого расчета в FF строится граф планирования. Кроме того, в отличие от HSP, использовалась другая поисковая стратегия, являющаяся комбинацией поиска экстремума и систематического алгоритма поиска (например, можно использовать обход в ширину). В FF использовались также дополнительные эвристики, позволявшие иногда сократить пространство поиска.

HAP (Vrakas, Tsoumakas, Bassiliades, Vlahavas, 2003)

HAP (Highly Adjustable Planner) или HAPRC — домено-независимый планировщик, осуществляющий поиск в пространстве состояний. Использует методы машинного обучения.

 ${\rm HAPNN}-{\rm модификация}$ планировщика HAP с другим алгоритмом машинного обучения.

BLACKBOX

BLACKBOX — планировщик, вобравший в себя идеи GRAPHPLAN и SATPLAN. Сначала строится граф планирования, на основании которого задача планирования переформулируется в булевскую задачу выполнимости. После этого решается именно задача выполнимости и затем из решения извлекается план (как в SATPLAN). Реализован гибкий подход к решению задачи выполнимости — в процессе поиска решения поочередно могут пробоваться разные методы. Это дает возможность планировщику решать широкий спектр задач.

DPPlan (Marcugini, Milani)

DPPlan — вариант планировщика Graphplan, осуществляющий поиск решения в графе путем упрощения его при помощи логических правил вывода.

Paragraph (Little, Thiebaux)

Paragraph — вероятностный планировщик, основанный на методологии GRAPHPLAN.

INTEGRA.NM(1996-97 гг.)

Первая версия технологии под торговой маркой ФинПлан была разработана в 1996–97 годах для решения проблем, связанных с планированием районного муниципального бюджета и контролированием его исполнения. Опыт использования этой системы показал, что с ее помощью можно эффективно решать задачи планирования, содержащие сотни параметров, связанных тысячами ограничений. В 1996 году система INTEGRA.NM была с успехом представлена на крупнейшей европейской компьютерной выставке CeBIT (Германия), а в 1997 году — демонстрировалась на международной конференции, посвященной практическим приложениям технологий, основанных на ограничениях (Англия).

За истекший период было последовательно реализовано несколько версий технологии. В частности, в 2004 году были существенно расширены пользовательские

и вычислительные возможности системы INTEGRA.NM и проведена ее адаптация для ряда приложений.

Одновременно технология INTEGRA.NM успешно опробована в рамках нескольких проектов Минобороны РФ, при разработке моделей промышленности Москвы, Томской и Ивановской областей, экспериментальных моделей экономики Республик Казахстан и Болгария, и в ряде других проектов. В настоящее время в рамках программы российско-белорусского научного сотрудничества РФФИ — БРФФИ по гранту №08-01-90018 (Ф08Р-019) на базе INTEGRA.NM ведется совместный проект «Разработка экспериментальной модели национальной экономики Республики Беларусь на базе недоопределенной информации с использованием методов программирования в ограничениях в рамках технологии нового поколения».

В ближайшие планы включена разработка многопользовательской и многотабличной версии INTEGRA.NM 3 на базе ядра UniCalc 5.0. Дальнейшее развитие технологии INTEGRA.NM будет идти в направлении ее дальнейшего расширения, в частности интеграции с технологией гибкого календарно-ресурсного планирования, развиваемой в проекте Time-EX.

Вычислительные возможности технологии INTEGRA.NM базируются на методе недоопределенных моделей (H-моделей) — оригинальном математическом и программном аппарате, который был разработан в нашем коллективе еще в начале 80-х годов. Этот аппарат превосходит по многим параметрам аналогичные западные разработки, которые стали появляться на программном рынке в последние несколько лет, и объединены общим термином — constraint programming (программирование в ограничениях). Данное направление программирования зародилось в рамках искусственного интеллекта и в настоящее время признается одним из самых перспективных методов решения сложных задач. Метод Н-моделей был представлен на десятках престижных международных конференций, где был оценен как один из наиболее мощных и развитых подходов в области программирования в ограничениях.

Semp-T

2001 год. Начата разработка программной среды SemP-N, которая будет отличаться от программной среды Semp-TAO повышенной надежностью и эффективностью, более высоким уровнем конструирования и отладки прикладных интеллектуальных систем, наличием интерфейса со стандартными базами данных, возможностью подключения внешних программных компонентов. Эта версия программной среды может рассматриваться, с одной стороны, как средство построения традиционных систем, основанных на знаниях, а с другой — как эффективная инструментальная система программирования в ограничениях.

Объединяет уникальный по мощности комплекс средств и методов представления и обработки знаний, включающий:

- высокоуровневые средства задания семантики объектов предметной области путем спецификации ограничений на значения их параметров и локальных правил вывода;
- иерархическую семантическую сеть с определяемыми свойствами отношений;

- аппарат для работы с неточно заданными (недоопределенными) значениями числовых, символьных и множественных типов;
- динамические типы данных;
- развитый аппарат продукционных правил с двумя уровнями средств динамического управления;
- средства генерации и проверки гипотез;
- объектную графику и высокоуровневые средства создания пользовательских интерфейсов;
- визуальный интерфейс разработчика.

Semp-T носит универсальный характер и может использоваться в любых сферах приложений, в которых требуется описание сложных по структуре и семантике предметных областей. Semp-T естественно сочетает мощный логический вывод и вычисления над неточно заданными параметрами, что обеспечивает ее эффективное применение в таких областях, как:

- экспертные системы и их проблемно-ориентированные оболочки;
- интеллектуальные базы данных и знаний;
- сложные диагностические системы;
- системы планирования и принятия решений;
- моделирование процессов в технике, экономике, биологии и социологии;
- интеллектуальные системы управления сложными объектами, в том числе роботами;
- компьютерная поддержка учебных курсов «Искусственный интеллект», «Инженерия знаний» и др.

Технология Semp-T ориентирована на конструктора интеллектуальных систем. Обеспечивает значительное повышение качества и многократное сокращение трудозатрат при создании сложных систем обработки знаний.

Технология Активных Объектов

ТАО — новая программная технология, обеспечивающая дальнейшее развитие объектно-ориентированного подхода на базе управления по данным. Получающая в мире все более широкое развитие мультиагентная архитектура существенно упрощает спецификацию поведения объектов, превращая их в активные автономные сущности, асинхронно взаимодействующие друг с другом и окружающей средой. При разработке ТАО особое внимание уделяется возможности ее применения для систем управления технологическими процессами, САПР, систем со сложными интерактивными интерфейсами и др. Благодаря мультиагентной архитектуре на основе управления по данным ТАО программы обладают естественной высокой параллельностью, не требующей никаких дополнительных усилий со стороны разработчика или пользователя.

Time-EX(R)

В 1995 г. Тіте-ЕХ/DOS прошла тестирование в фирме SAIC (США), в рецензии которого было отмечено, что Тіте-ЕХ в сравнении с MS Project и Symantec TimeLine обладает рядом отличных качеств, которые могут быть действительными преимуществами для фирм, планирующих большие проекты в условиях неопределенности относительно сроков и наличия ресурсов.

В 1995 г. начата разработка прототипной версии, ориентированной на Windows 3.1.

Массовый переход на WIndows 95 привел к необходимости создания новой версии, которая была начата в 1997 г. и практически закончена в 1999-м, послужив основой очередной версии 2.2, разработка которой была запущена в начале 2000-го года. Эта версия имеет современную архитектуру, расширенную модель времени и развитый эргономичный интерактивный интерфейс.

Ресурсно-календарное планирование на основе неполных данных.

Интеллектуальная система планирования и управления проектами нового поколения Time-EX основана на оригинальной технологии обработки знаний, которая существенно превосходит методы, реализованные в самых современных коммерческих системах календарного планирования.

Рассчитывает и оптимизирует календарные планы при минимуме информации о временных параметрах работ проекта, используя неполные и приблизительные данные, типичные для повседневных ситуаций практической деятельности. Частично определенный график, который строится системой даже при минимуме исходных данных, уточняется и оптимизируется по мере поступления дополнительной информации

Гарантирует наилучшие сроки и экономические показатели календарного плана, обеспечивает необходимые резервы времени и динамически адаптируется к изменению условий реального хода выполнения.

Может использоваться в качестве ядра систем планирования и управления для проектов различного типа, финансовой деятельности, документооборота, мероприятий в чрезвычайных ситуациях и др.

При соответствующей специализации Time-EX может найти эффективное использование в планировании персональной деятельности, планировании транспортных потоков, управлении производственными процессами, и т.п. Системе обеспечено широкое применение на всех уровнях муниципального управления, а также в деятельности предприятий и организаций любого типа и размера.

Ведется разработка следующей существенно расширенной версии системы.

Неосистемы: Технико-экономическое планирование (2007 г.)

Программный продукт позволяет моделировать деятельность предприятия путем формирования целостной системы операционных и финансовых планов, расчета и анализа плановых показателей.

Реализованы следующие функции:

- формирование портфеля заказов, бюджета продаж в натуральном и стоимостном выражении;
- планирование и формирование графиков производства в натуральном выражении;

- расчет и планирование потребностей в материалах и комплектующих с учетом графиков производства в натуральном выражении;
- планирование закупок, поступления ресурсов на склады предприятия в натуральном и стоимостном выражении;
- планирование расходов, расчет себестоимости продукции и незавершенного производства;
- планирование начисления, возмещения и уплаты НДС.



Контрольные вопросы и задания к главе 6

- 1) Определите понятие «планирование» в интеллектуальных системах.
- 2) Опишите помеченный неправильный граф, называемый планом.
- 3) Дайте характеристику двух миров (сред), классического планирования и планирование в реальном мире.
- 4) Расскажите о планировании с помощью поиска в пространстве состояний.
- 5) Перечислите и охарактеризуйте сложные и неправильные методы поиска пути в пространстве состояний.
- 6) Поясните планирование с частичным упорядочиванием.
- 7) Поясните планирование с помощью пропозиционной логики.
- 8) Сформулируйте процесс планирования иерархической сети задач.
- 9) Проанализируйте планирование и осуществление действий в реальном мире.
- 10) Перечислите четыре метода планирования для осуществления действий в условиях недетерминированности.
- 11) Поясните планирование без использования датчиков.
- 12) Для чего необходимы контроль выполнения плана и перепланирования?
- 13) Что понимают под условным иссанированием и непрерывным планированием?
- 14) Перечислите методы решения задач планирования.
- 15) Опишите метод ключевых состояний и ключевых операторов.
- 16) Представьте графически основные методы механизма редукции задачи.
- 17) Приведите пример работы механизма редукции задачи.
- 18) Перечислите методы планирования с помощью логического вывода.
- 19) Что представляет собой система QA3.
- 20) Назовите характерные особенности методов продукций системы STRIPS и использующие макрооператоры.

- 21) Рассмотрите планирующую систему STRIPS.
- 22) Представьте схему работы планирующей системы STRIPS.
- 23) Рассмотрите реализацию алгоритмов планирования и приведите пример.
- 24) Дайте краткий анализ систем интеллектуального планирования.



Литература к главе 6

[1] Попов Э. В. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта / Э. В. Попов, Г. Р. Фирдман. — М. : Изд-ва «Наука», 1976.-456 с.

......

- [2] Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин [и др.]; под ред. В. Н. Вагина и Д. А. Поспелова. М.: Физматлит, 2004. 704 с.
- [3] Искусственный интеллект. Применение в интегрированных производственных системах / под ред. Э. Кюснаке. М.: Машиностроение, 1991.
- [4] Лорьер Ж. Л. Системы искусственного интеллекта : пер с франц. В. Л. Стефанюка / Ж. Л. Лорьер. М. : Мир. 1991. 568 с.
- [5] Нильсон Н. Искусственный интеллект. Методы поиска решений : пер. с англ. / Н. Нильсон. М. : Радио и связь, 1985. 280 с.
- [6] Поспелов Д. А. Семиотические модели в задачах планирования для систем искусственного интеллекта / Д. А. Поспелов, Е. И. Ефимов // Изв. АН СССР. Сер. Техническая кибернетика. 1977. №6. С. 60–68.
- [7] Рассел С. Искусственный интеллект: современный подход: пер. с англ. К. А. Итицына. — 2-е изд. / С. Рассел, П. Норвиг. — М.: Вильямс. 2006. — 1408 с.
- [8] Черноруцкий И. Г. Методы принятия решений / И. Г. Черноруцкий. СПб. : БХВ Петербург, 2005. 416 с.
- [9] Newell A. GPS: A Program that Simulates Human Thought / A. Newel, H. Simon // Computers and Thought. −1963. −№4. −P. 279–293.

Глава 7

ЭКСПЕРТНЫЕ СИСТЕМЫ

7.1 Классификация экспертных систем

В рамках научного направления «искусственный интеллект» в настоящее время происходит бурное исследование, разработка и внедрение новых информационных технологий, то есть создание новых систем обработки информации. Научный и коммерческий успех в мире в середине 80-х — начале 90-х годов экспертных систем (ЭС) — самого развивающегося направления искусственного интеллекта показал, что в реальной практике существует большой класс задач, которые не удается решать методами традиционного программирования (формирование решения задачи в математических терминах). К ним относятся задачи, не имеющие алгоритмического решения, то есть неформализованные задачи со следующими особенными характеристиками [11, 15]: они не могут быть заданными в числовой форме; цели не могут быть выражены в терминах точно определенной целевой функции; алгоритмическое решение если и существует, то его нельзя использовать из-за ограниченности ресурсов.

Таким образом, алгоритм как фундаментальное понятие в вычислительной математике и в программной технологии в целом, чье доминирующее положение было изначально предопределено самой эпохой компьютеризации, значительно уступил свои позиции понятию «модель», поскольку основная парадигма искусственного интеллекта (в отличие от математики) ориентирована не на разработку формальных аппаратов, а на выявление и изучение различных форм знаний и только потом на создание комплекса формальных средств, описывающих способы репрезентации и манипулирования знаниями [10, 11]. Модель и возможность прямого взаимодействия с ней стали основным базисом, как для неформальных задач, так и в традиционном программировании в связи с развитием объектноориентированного подхода (ООП) и технологии объектной модели компонентов (СОМ — Сотропент Object Model), хотя здесь остается алгоритмический характер управления процессом программы. Работы по разработке и внедрению ЭС в нашей

стране ведутся 15–20 лет, и уже накоплен определенный опыт их использования. Этапы создания отечественных разработок ЭС и сравнительный анализ их характеристик представлен в таблице 7.1 С. Г. Чикиновым [16].

Таблица 7.1 – Характеристики используемых и перспективных ЭС.

Основные ха-	Экспертные	Экспертные систе-	Перспективные
рактеристики	системы пер-	мы второго поколе-	ЭС 2000–2010 гг.
рактористики	вого поколения	ния (1996–2000 гг.)	(интеллектуальные
	(1985–1996 гг.)	11111 (1990 2000 11.)	интегрированные
	(1703 1770 11.)		инструментальные
			средства)
Carrett	414	(Managaran arra	* '
Способы из-	«Инженер зна-	«Инженер зна-	Автоматическое об-
влечения зна-	ний», эмпири-	ний», исследова-	наружение из баз
ний, структура	ческие знания	ния по автоматиче-	знаний (из текстов,
баз знаний	эксперта	скому извлечению	руководств, инструк-
		знаний из базы	ций, схем и.т. д.)
		знаний	
Типы баз зна-	Отдельные	Работа с любыми	Библиотека форм зна-
ний	формы —	формами зна-	ний, имитационные
	продукция,	ний (библиотека	модели, сценарии
	фреймы, се-	знаний)	
	мантические		
	сети, решаю-		
	щие деревья,		
Источник зна-	Эксперт	Эксперты, базы	Базы знаний, данные,
ний		знаний	статистические или
			эмпирические табли-
			цы триад объектов
			«объект — свойство —
			время»
Наличие базы	Отсутствует	Частично, базы	Базы данных из
данных		данных из двух-	трехвходовых таблиц
		кодовых таблиц	«объект — свойство —
		«объект-свойство»	время»
Логический	Вывод по де-	Дедукция, нечет-	Дедукция, индукция,
вывод	дукции	кие выводы,	немонотонные рас-
		индукция, немо-	суждения, методы
		нотонные рассуж-	близости в про-
		дения, частично	странстве знаний,
		рассуждения по	рассуждения по
		аналогии	аналогии
продолжение на следующей странице			

Таблица 7.1 – Характеристики используемых и перспективных ЭС.

Основные ха-	Эконования	Эканартина аната	Поположения
	Экспертные	Экспертные систе-	Перспективные ЭС 2000–2010 гг.
рактеристики	системы пер-	мы второго поколе-	
	вого поколения	ния (1996–2000 гг.)	(интеллектуальные
	(1985–1996 гг.)		интегрированные
			инструментальные
			средства)
Язык общения	Фразы и тер-	Проблемно-	Сценарии диало-
с пользовате-	мины жесткой	ориентированный	га, формирование
лем	конструкции	естественный язык	терминологии под
	прикладной		прикладную область
	области		и форм сообщений,
			понимание смысла
			сообщений
Устный диалог	Отсутствует	Ограниченный	Словари по опреде-
		словарь	ляющим терминам
		• • • • • • • • • • • • • • • • • • •	прикладных областей
			в процессе общения
			и использования
Проверка адек-	Отсутствует	Частичная провер-	Проверка непротиво-
ватности баз	Отсутствуст	ка непротиворечи-	речивости, полноты,
знаний		вости и полноты	работа с информаци-
эпапии			ей с НЕ-факторами
		баз знаний	
			(неопределенность,
			неполнота, неточ-
П		11	ность)
Прогнозирова-	Отсутствует	Исследования по	Автоматическое
ние недоста-		автоматическому	прогнозирование
ющих данных		прогнозированию	величин, отсутствую-
в базе данных		величин, отсут-	щих в базе данных
		ствующих в базе	
		данных	
Модель поль-	Отсутствует	Обучение систем	Программы адапта-
зователя		по адаптации	ции под конкретных
		под конкретного	пользователей и ин-
		пользователя	терфейс с АСУ
Средства ав-	Отсутствует	Частично обеспе-	Программы гомеоста-
томатического		чивают автомати-	та для определения
обеспечения		зацию в некоторых	работоспособности
работоспособ-		областях	системы, обнаруже-
ности системы			ния новых закономер-
			ностей и индикации
			сведений для пользо-
			вателя
	продолжение на следующей странице		
-F-Comment of the company of the com			

Таблица 7.1 – Характеристики используемых и перспективных ЭС.

Основные ха-	Экспертные	Экспертные систе-	Перспективные
рактеристики	системы пер-	мы второго поколе-	ЭС 2000–2010 гг.
P	вого поколения	ния (1996–2000 гг.)	(интеллектуальные
	(1985–1996 гг.)	(1990 2000 111)	интегрированные
	(1500 1550 111)		инструментальные
			средства)
Выдаваемые	Числовые дан-	Числовые данные,	Числовые дан-
результаты	ные, стандарт-	рекомендации	ные, рекомендации
результаты	ные рекомен-	с сопровождаю-	с сопровождаю-
	дации	щим пояснением,	щим пояснением,
	диции	обучение систем	формулирование об-
		обнаружению	наруженных новых
		новых закономер-	закономерностей,
		ностей	тенденций, графика,
		1001011	анимация, диаграммы
Вид обрабаты-	Статическая	Динамическая,	Статическая, динами-
ваемой инфор-		статическая	ческая, потоковая, ди-
мации			намическая в АСУ
Объем	Сотни правил	Тысячи правил	Десятки тысяч правил
эвристических	T.	F	
знаний			
в базе знаний			
Принцип	Обособленное	Гибридное постро-	Гибридные интеллек-
построения	использование	ение ЭС (эвристи-	туальные нечеткие
и использова-	ЭС для реше-	ческая и имитаци-	системы (интеллек-
ния системы	ния задачи	онная компоненты)	туальные интегриро-
		,	ванные комплексы
			моделирования),
			открытая система
Обработка рас-	Отсутствует	Исследования	Распределение ЭС,
пределенных		по построению	многоагентные систе-
знаний		распределенных	мы искусственного
		ЭС, разработка	интеллекта
		концепции	
Функции	Пассивный по-	Активный помощ-	Управляющий орган
системы	мощник поль-	ник пользователя,	исполнительными
	зователя	обработка на стен-	механизмами в АСУ,
		дах функций ЭС по	активный помощник
		управлению испол-	пользователя
		нительными меха-	
		низмами в АСУ	
продолжение на следующей странице			

Основные ха-	Экспертные	Экспертные систе-	Перспективные
рактеристики	системы пер-	мы второго поколе-	ЭС 2000–2010 гг.
	вого поколения	ния (1996–2000 гг.)	(интеллектуальные
	(1985–1996 гг.)		интегрированные
			инструментальные
			средства)
Ввод новых	В режиме	В рабочем режиме	В рабочем режиме ЭС
знаний, мо-	ввода инфор-	ЭС	и АСУ
дификация	мации		
знаний			
Запуск меха-	По требова-	По требованиям,	Автоматически, при
низма вывода	нию пользова-	автоматически при	смене ситуационных
решения	теля	смене входных	исходных данных,
		данных	в том числе в АСУ
Объективизация	Отсутствует	Частично (в стадии	Методы нечеткого
субъективных		исследований)	адаптивного управле-
данных			Р В В В В В В В В В В

Таблица 7.1 – Характеристики используемых и перспективных ЭС.

Вывод решений в простой традиционной ЭС производится путем сопоставления с образцом фактов из базы текущих данных и правил, иначе говоря, механизм выполнения обычных программ здесь заменялся машиной вывода, автономной к проблемной области. Таким образом, развитие программирования позволило перейти к имитации интеллектуальной деятельности человека, позволяющей создавать программные решения неформализованных задач, причем на этом этапе стало осуществляться слияние данных с алгоритмами (программами) и создаваться базы знаний (БЗ), то есть модели проблемных областей. ЭС первого поколения нашли широкое применение как автономные программные системы во многих областях, традиционно относящихся к слабо- и средне-структурированными. Тенденции к интеграции парадигмы ИИ с другими научно-техническими парадигмами (подходы и методы математической статистики, имитационное моделирование, технология программирования и.т. д.) при создании приложений привели к конструированию различных гибридных систем.

Таким образом, концепция разработки традиционных (простых) ЭС, декларированная в конце 80-х годов [4, 11], ориентировалась на изолированное автономное функционирование ЭС, построенной только на основе формальных и эвристических знаний экспертов, описывающих ограниченную проблемную область, модель которой создавалась вручную за счет искусной работы инженера по знаниям с экспертом [1–3]. В реальности оказалось, что объемы информации, использующейся для работы интегрированных или гибридных ЭС, должны быть намного больше, и без применения специальных средств автоматизации процессов приобретения и верификации знаний, обработки недостоверных знаний (то есть знаний, содержащих неопределенность, неточность, нечеткость, неполноту и другие виды не-

факторов) построение и программное сопровождение моделей проблемных областей является почти непосильной задачей для разработчиков.

С другой стороны, предлагавшиеся инструментальные средства для построения традиционных ЭС в виде различных оболочек не поддерживали какой-либо четкой методологии и технологии анализа, проектирования и разработки прикладных ЭС (по типу современных CASE-систем), что делало практически невозможным использование их для построения систем с более сложной архитектурой типа интегрированных ЭС.

Значимость ЭС первого поколения заключается в том, что в них отработаны теоретические основы искусственного интеллекта и экспертных систем, методы внутреннего представления внешнего мира и логического вывода решений. Приобретенный опыт по конкретизации и углублению накопленных знаний явился необходимым трамплином для массового развертывания дальнейших работ по прикладному применению средств и методов искусственного интеллекта.

7.2 Экспертные системы первого и второго поколений

Экспертные системы первого поколения. Экспертные системы первого поколения (1985–1996 гг.) в основном содержат исследовательские прототипы для исследования и обоснования теоретических основ искусственного интеллекта. Проводившиеся в этот период исследования носили фундаментальный характер, направленный на исследование отдельных фрагментов приобретения, представления и использования знаний, различных механизмов вывода. По каждому фрагменту было создано множество исследовательских прототипов. Исследования проводились для статичных условий, где во многих ситуациях наблюдался «тривиальный эффект», а сами ЭС являлись статическими системами [11].

ЭС разрабатывались для автономного применения. В результате был накоплен солидный багаж методов и инструментальных средств, которые составили основу искусственного интеллекта как научной дисциплины. Однако многообразие задач управления, для решения которых использовались вычислительные средства, и их специфичность не позволили говорить о создании какой-либо универсальной технологии.

Экспертные системы первого поколения представляли собой системы с интеллектом пассивного ассистента пользователя: они располагали только теми знаниями, которые были получены от экспертов, переработаны «инженерами знаний» и введены в память системы (базу знаний) в удобном для машины виде. Система была способна манипулировать этими знаниями, имитируя процесс логического вывода, и выдавать ответы на запросы пользователя. Система не имела механизмов, которые позволяли бы ей критически оценивать вводимые в ее память знания, выявлять в них противоречия, автоматически обнаруживать закономерности, использовать их для предсказания и извлекать новые знания из данных [5].

Экспертные системы второго поколения—динамические системы реального времени. Динамические ЭС (экспертные системы реального времени—ЭС РВ) составили основное направление работ по созданию ЭС второго поколения

(1996–2000 гг.) и содержат качественное изменение ЭС. В их архитектуру введены новые компоненты — подсистемы моделирования внешнего мира, связи с внешним окружением, учета временной логики обрабатываемых событий. Эти системы обладают более удобными средствами взаимодействия. Они стали «активными» ЭС, выполняющими роль активного помощника пользователя (партнерские системы) или исполнительного механизма в автоматизированных системах управления (управленческие ЭС).

Технология построения современных АСУ уже предусматривает обработку комплекса взаимоувязанных задач и потоков информации обрабатываемых процессов что определило применение в них динамических ЭС, обрабатывающих динамическую обстановку. Это потребовало более глубокой проработки методологических вопросов создания и применения ЭС. В динамических системах результаты измерений множества параметров потоковой информации обрабатываются и выдаются оператору в виде информации о ходе функционирования АСУ, меньшей по объему, но более содержательной [17].

ЭС РВ должны обеспечивать выполнение временного интервала, в течение которого реакция ЭС на входное воздействие не превышает заданного значения (1 %, 5 % и.т. д.), определяемого необходимой точностью моделирования конкретной АСУ (так называемого показателя постоянной реального времени — Π BP)

Данные системы обладают средствами самостоятельного извлечения знаний из данных, поступающих в систему в ходе ее создания и эксплуатации. На этом основании системы второго поколения способны обнаруживать противоречия между имеющимися и вновь поступающими знаниями и данными (проверка адекватности базы знаний) и частично обладают средством извлечения, представления и обработки нечетких и неопределенных знаний (так называемых знаний с нефакторами).

Обработка нечетких знаний и создание нечетких экспертных систем составляют основное содержание работ в области экспертных систем в настоящее время. Область управления явилась одной из предпосылок возникновения идеи нечетких множеств, которая помогла решить проблему надежности в экспертных системах. Почти все ЭС управления обеспечивают нечеткое управление, т. е. являются нечеткими ЭС. В области практического применения число нечетких ЭС уже значительно превысило число традиционных ЭС.

Основным отличием ЭС PB от экспертных систем первого поколения является интегрированность (рис. 7.1).

7.3 Классификация ИЭС, взаимосвязь процессов интеграции и гибридизации в ИЭС

Как показано на рис. 7.1, сегодня в ИИ происходят два основных процесса, один из которых связан с расширением архитектур интеллектуальных систем за счет добавления новых инструментов из других областей знаний (интеграционный процесс), а другой—с извлечением новых подобластей из хорошо известных и созданием для них специальных инструментов (разделительный процесс).

Поскольку ИЭС – это программная система, в архитектуре которой наряду

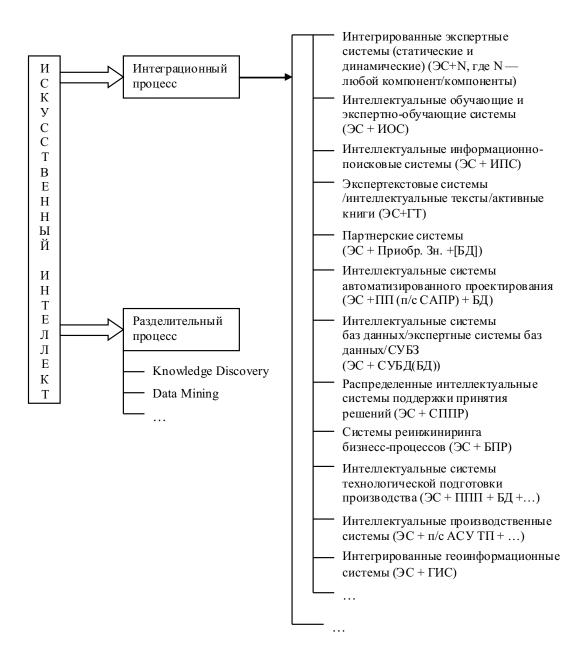


Рис. 7.1 – Примеры интегрированных экспертных систем.

с традиционным компонентом «ЭС», использующим для решения НФ-задач, как правило, методологию простых продукционных ЭС, содержатся некоторые компоненты «N», расширяющие функциональные возможности ЭС (например, БД, пакеты прикладных программ, обучающие системы и т. д.), то все ИЭС целесообразно разделить на два подкласса [13, 14]: ИЭС с поверхностной интеграцией компонентов и ИЭС с глубинной интеграцией компонентов (рис. 7.2).

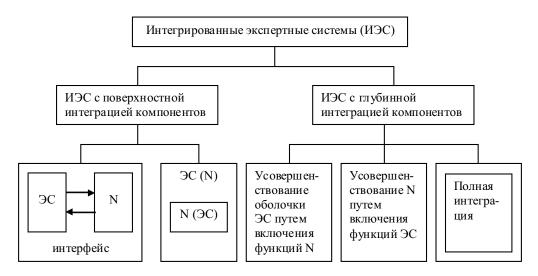


Рис. 7.2 – Классификация ИЭС, где N — любая программная система (например, СУБД, ППП и т. п.).

Тогда в случае поверхностной интеграции компонентов ЭС и N взаимодействие, например, СУБД и ЭС может осуществляться на уровне передачи сообщений, причем результаты работы каждого компонента программная система (например, СУБД, ППП и т. п.) исходные данные для другого. Более сложным вариантом поверхностной интеграции является такой, в котором один из его программных компонентов (ЭС или N) в ходе своего функционирования обращается для уточнения данных или решения каких-то локальных задач к другому компоненту. В этом случае реализуется более тесное сопряжение между ЭС и N, например с помощью использования специальных программ, служащих «мостом» между СУБД и ЭС для передач сообщений.

Под глубинной интеграцией компонентов ЭС и N понимается усовершенствование этих компонентов путем приобретения ими функций, несвойственных традиционным типам этих компонентов. Например, решение вопроса интеграции за счет усовершенствования используемой оболочки ЭС путем включения отдельных функций СУБД ставит своей целью встроить обобщенную СУБД внутрь базы знаний (БЗ) [7]. Данный подход представляет наибольший интерес в том случае, если БЗ слишком велика, чтобы ее можно было разместить в памяти машины, и когда требуются эффективные механизмы доступа к данным. С другой стороны, говоря об усовершенствовании СУБД путем включения базовых функций ЭС, необходимо отметить, что СУБД обладает большим специализированным механизмом для выполнения таких задач, как хранение данных, поддержание целостности данных, навигация по БД, эффективное выполнение сложных запросов. При этом данный механизм может быть насыщен также функциями ЭС, в частности, один

из подходов к решению этой задачи заключается в расширении СУБД с целью ее преобразования в обобщенный механизм вывода [7]. Следует отметить, что в случае проектирования ИЭС с глубинной интеграцией компонентов более предпочтительным является подход, связанный с усовершенствованием инструментальных средств для ЭС путем включения нетрадиционных для них функций, что было, в частности, использовано при разработке задачно-ориентированной методологии построения ИЭС.

Полная интеграция является самым высоким уровнем интеграции для ИЭС и заключается в соединении лучших качеств компонентов ЭС и N. Например, полная интеграция СУБД и ЭС заключается в отборе лучших особенностей и механизмов построения СУБД и преимуществ ЭС, в определении новых моделей данных и знаний, в создании совершенно новых систем с новыми возможностями. Бесспорные преимущества этого подхода к интеграции компонентов ЭС и СУБД заключаются в том, что все компоненты находятся в одной системе с унифицированной структурой для моделирования фактов и правил и однородной обработкой данных и знаний.

Анализировать тенденции к интеграции и гибридизации в современных ИЭС существенно помогает так называемая «многоуровневая модель процессов интеграции», предложенная в рамках задачно-ориентированной методологии, в соответствии с которой процессы интеграции предлагается рассматривать с точки зрения следующих аспектов [14].

Интеграция в архитектуре ИЭС различных компонентов, реализующих как Фзадачи (компонент N), так и НФ-задачи (компонент ЭС) и определяющих специфику функционирования ИЭС в целом (верхний уровень интеграции).

Интеграция (функциональная, структурная, концептуальная), связанная с используемыми концепциями и методологиями проектирования, разработки конкретных классов ИЭС и их компонентов (средний уровень интеграции).

Интеграция (информационная, программная, техническая), связанная с используемыми технологиями, инструментальными средствами и платформами (нижний уровень интеграции).

Подобная модель позволяет, как минимум для трех уровней интеграции в ИЭС, четко проследить взаимосвязь конкретных способов интеграции отдельных компонентов ИЭС (N или ЭС) с необходимостью внутреннего сращивания, т.е. гибридизации отдельных моделей, методов, процедур, алгоритмов и т. д. На рис. 7.3 для ИЭС с поверхностной и глубинной интеграцией компонентов N и ЭС детально проиллюстрированы используемые на практике подходы и степень необходимой гибридизации, в соответствии с чем сделаны следующие выводы и предложения [14, 15]:

- 1) Понятие «интегрированная интеллектуальная система» не синонимично понятию «гибридная интеллектуальная система» и является более широким, чем понятие «гибридная интеллектуальная система».
- 2) Интегрированная интеллектуальная система не обязательно гибридная, и наоборот.
- 3) Интегрированная интеллектуальная система всегда гибридная только в случае полной интеграции компонентов.

Способы интеграции (верхний уровень)	Необходимость гибридизации	
1. ИЭС с поверхностной интеграцией компонентов	Отсутствует (используются автономные модели, методы и алгоритмы)	
2. ИЭС с глубинной интеграцией компонентов 2.1. ЭС Функции N	Слабая (используются в основном автономные модели и методы с частичной модификацией отдельных моделей, методов и алгоритмов)	
2.2. N Функции ЭС	Слабая (используются в основном автономные модели и методы с частичной модификацией отдельных моделей, методов и алгоритмов)	
2.3. (ЭС) N Полная интеграция — Селекция лучших особенностей, функций и механизмов ЭС и N — Определение новых моделей данных и знаний — Построение ИЭС с новыми функциями и возможностями …	Сильная: Разрабатывается единая унифицированная структура для моделирования фактов и правил; присугствует однородная обработка данных и знаний; все компоненты (модули) нах одятся в одной системе и т.д.	

Рис. 7.3 – Взаимосвязь способов интеграции компонентов ИЭС с необходимостью гибридизации.

Поэтому термин «интегрированные экспертные системы» необходимо использовать для сложных программных систем, объединяющих методы и средства ЭС с технологиями традиционного программирования. Термин «гибридные экспертные системы» целесообразно применять как для ЭС, так и для инструментальных средств (ИС) для ЭС с гибридными способами представления знаний (т. е. объединяющих, например, внутри одного ИС различные модели представления знаний и механизмы функционирования).

7.4 Структура и компоненты экспертных систем

Типовая архитектура ЭС предоставлена на рис. 7.4. Взаимодействие с ЭС осуществляется с помощью *интерфейса*. В качестве пользователя ЭС выступает либо конечный пользователь (лицо принимающее решение), либо эксперт, либо инженер по знаниям. Этим категориям пользователей предоставляются разные возможности взаимодействия с системой. В ЭС применяются различные варианты реализа-

ции интерфейса: меню-ориентированный, графический, командный, естественно-языковый, речевой. Кроме обеспечения взаимодействия с различными категориями пользователей, интерфейс выполняет необходимые функции сопряжения с внешними объектами: базами данных, различными датчиками, коммуникационным оборудованием и т. д.

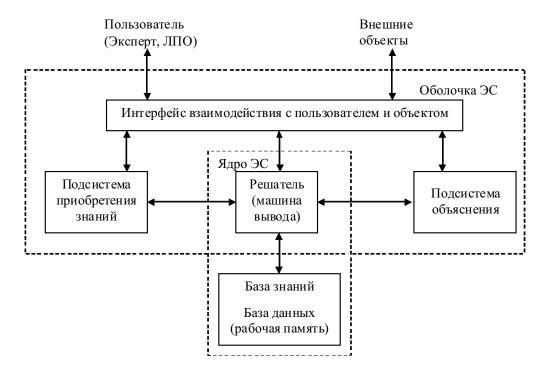


Рис. 7.4 – Структурная схема ЭС.

 $\mathcal{A}\partial po\ \mathcal{C}$ образует базу данных, базу знаний и машину выводов. База знаний представляет собой рабочую память, в которой хранятся текущие данные, заключения и другая информация, имеющая отношение к анализируемой системой ситуации. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах (ИПС) и в системах управления базами данных (СУБД) для обозначения всех данных (в первую очередь долгосрочных), хранимых в системе.

База знаний (БЗ) в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области. База знаний обеспечивает хранение знаний, представленных с помощью одной из моделей: логической, продукционной, фреймовой, сетевой.

Решатель (машина вывода, подсистема поиска решений), используя исходные данные из рабочей памяти и знания из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи в соответствии с используемой моделью знаний. Например, при использовании продукционной модели представления знаний выполняется цикл распознавание-действие. Процедуры, реализующие цикл управления, отделены от продукционных знаний. Отделение знаний, хранящихся в ЭС, от алгоритмов поиска решений позволяет:

- представлять знания в более естественной форме по сравнению со знаниями, встроенными непосредственно в процедуры;
- разработчику ЭС в большей степени сосредоточиться на накоплении и организации предметных знаний, а не на деталях компьютерной реализации;
- изменять содержимое баз знаний без необходимости внесения изменений в программный код;
- создавать оболочки ЭС, которые можно наполнять знаниями различных предметных областей.

Подсистема объяснения ЭС позволяет пользователю выяснить или объяснить, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала. Это облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

Подсистемы (компонент) приобретения знаний используются как с целью автоматизации процесса наполнения ЭС знаниями, так и при корректировке базы знаний, при ее обновлении, пополнении или исключении элементов знаний. Данные процессы реализуются либо при непосредственном участии инженера по знаниям (эксперта), либо в автоматическом режиме в ходе обучения системы.

В разработке ЭС участвуют представители следующих специальностей:

- 1) эксперт в проблемной области, задачи которой будет решать ЭС;
- 2) инженер по знаниям специалист по разработке ЭС (используемые им технологии, методы называют технологиями (методами) инженерии знаний);
- 3) программист по разработке инструментальных средств (ИС), предназначенных для ускорения разработки ЭС.

Необходимо отметить, что отсутствие среди участников разработки инженеров по знаниям (т. е. их замена программистами) либо приводит к неудаче процесс создания ЭС, либо значительно удлиняет его.

Эксперт определяет знания (данные и правила), характеризующие предметную область, обеспечивает полноту и правильность введения в ЭС знаний

Инженер по знаниям помогает эксперту выявить и структурировать знания, необходимые для работы ЭС; осуществить выбор того ИС, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний в этом ИС; выделяет и программирует (традиционными средствами) стандартные функции (типичные для данной проблемной области), которые будут использоваться в правилах, вводимых экспертом.

Программист разрабатывает ИС (если ИС разрабатывается заново), содержащую в пределе все основные компоненты ЭС, и осуществляет его сопряжение с той средой, в которой оно будет использовано.

Экспертная система работает в двух режимах: режим приобретения знаний и режим решения задачи (называемый также режимом консультации или режимом использования ЭС).

В режиме приобретения знаний общение с ЭС осуществляет (через посредничество инженера по знаниям) эксперт. В этом режиме эксперт, используя компонент приобретения знаний, наполняет систему знаниями, которые позволяют ЭС в режиме решения самостоятельно (без эксперта) решать задачи из проблемной об-

ласти. Эксперт описывает проблемную область в виде совокупности данных и правил. Данные определяют объекты, их характеристики и значения, существующие в области экспертизы. Правила определяют способы манипулирования с данными, характерные для рассматриваемой области.

Отметим, что режиму приобретения знаний в традиционном подходе к разработке программ соответствуют этапы алгоритмизации, программирования и отладки, выполняемые программистом. Таким образом, в отличие от традиционного подхода в случае с ЭС разработку программ осуществляет не программист, а эксперт (с помощью ЭС), не владеющий программированием.

В режиме консультации общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ его получения. Необходимо отметить, что в зависимости от назначения ЭС пользователь может не быть специалистом в данной проблемной области (в этом случае он обращается к ЭС за результатом, не умея получить его сам) или быть специалистом (в этом случае пользователь может сам получить результат, но он обращается к ЭС с целью либо ускорить процесс получения результата, либо возложить на ЭС рутинную работу). В режиме консультации данные о задаче пользователя после обработки их диалоговым компонентом поступают в рабочую память. Решатель на основе входных данных из рабочей памяти, общих данных о проблемной области и правил из БЗ формирует решение задачи. ЭС при решении задачи не только исполняет предписанную последовательность операций, но и предварительно формирует ее. Если реакция системы не понятна пользователю, то он может потребовать объяснения.

7.5 Этапы разработки экспертных систем

Разработка ЭС имеет существенные отличия от разработки обычного программного продукта. Опыт создания ЭС показал, что использование при их разработке методологии, принятой в традиционном программировании, либо чрезмерно затягивает процесс создания ЭС, либо вообще приводит к отрицательному результату.

Использовать ЭС следует только тогда, когда разработка ЭС возможна, оправдана и методы инженерии знаний соответствуют решаемой задаче. Чтобы разработка ЭС была возможной для данного приложения, необходимо одновременное выполнение, по крайней мере, следующих требований:

- 1) существуют эксперты в данной области, которые решают задачу значительно лучше, чем начинающие специалисты;
- 2) эксперты сходятся в оценке предполагаемого решения, иначе нельзя будет оценить качество разработанной ЭС;
- 3) эксперты способны вербализовать (выразить на естественном языке) и объяснить используемые ими методы, в противном случае трудно рассчитывать на то, что знания экспертов будут «извлечены» и вложены в ЭС;
- 4) решение задачи требует только рассуждений, а не действий;
- 5) задача не должна быть слишком трудной (т. е. ее решение должно занимать у эксперта несколько часов или дней, а не недель);
- 6) задача хотя и должна быть выражена в формальном виде, но все же должна

- относиться к достаточно «понятной» и структурированной области, т. е. должны быть выделены основные понятия, отношения и известные (хотя бы эксперту) способы получения решения задачи;
- 7) решение задачи не должно в значительной степени использовать «здравый смысл» (т. е. широкий спектр общих сведений о мире и о способе его функционирования, которые знает и умеет использовать любой нормальный человек), так как подобные знания пока не удается (в достаточном количестве) вложить в системы искусственного интеллекта.

Использование ЭС в данном приложении может быть возможно, но не оправдано. Применение ЭС может быть оправдано одним из следующих факторов:

- 1) решение задачи принесет значительный эффект, например экономический;
- 2) использование человека-эксперта возможно. Невозможность возникает либо из-за недостаточного количества экспертов, либо из-за необходимости выполнять экспертизу одновременно в различных местах;
- 3) использование целесообразно в тех случаях, когда при передаче информации эксперту происходит недопустимая потеря времени и информации;
- 4) использование ЭС целесообразно при необходимости решать задачу в окружении, враждебном для человека.

Приложение *соответствует* методам ЭС, если решаемая задача обладает совокупностью следующих характеристик:

- 1) задача может быть естественным образом решена посредством манипуляции с символами (т.е. с помощью символических рассуждений), а не манипуляции с числами, как принято в математических методах и в традиционном программировании;
- 2) задача должна иметь эвристическую, а не алгоритмическую природу, т. е. ее решение должно требовать применения эвристических правил. Задачи, которые могут быть гарантированно решены (с соблюдением заданных ограничений) с помощью некоторых формальных процедур, не подходят для применения ЭС;
- 3) задача должна быть достаточно сложна, чтобы оправдать затраты на разработку ЭС. Однако она не должна быть чрезмерно сложной (решение занимает у эксперта часы, а не недели), чтобы ЭС могла ее решать;
- 4) задача должна быть достаточно узкой, чтобы решаться методами инженерии знаний, и практически значимой.

При разработке ЭС, как правило, используется концепция «быстрого прототипа». Суть этой концепции состоит в том, что разработчики не пытаются сразу построить конечный продукт. На начальном этапе они создают прототип (прототипы) ЭС. Прототипы должны удовлетворять двум противоречивым требованиям: с одной стороны, они должны решать типичные задачи конкретного приложения, а с другой — время и трудоемкость их разработки должны быть весьма незначительны, чтобы можно было максимально запараллелить процесс накопления и отладки знаний (осуществляемый экспертом) с процессом выбора (разработки) программных средств (осуществляемым инженером по знаниям и программистом).

Для удовлетворения указанным требованиям, как правило, при создании прототипа используются разнообразные средства, ускоряющие процесс проектирования. Прототип должен демонстрировать пригодность методов инженерии знаний для данного приложения. В случае успеха эксперт с помощью инженера по знаниям расширяет знания прототипа о проблемной области. При неудаче может потребоваться разработка нового прототипа или разработчики могут прийти к выводу о непригодности методов ЭС для данного приложения. По мере увеличения знаний прототип может достигнуть такого состояния, когда он успешно решает все задачи данного приложения. Преобразование прототипа ЭС в конечный продукт обычно приводит к перепрограммированию ЭС на языках низкого уровня, обеспечивающих как увеличение быстродействия ЭС, так и уменьшение требуемой памяти. Трудоемкость и время создания ЭС в значительной степени зависят от типа используемого инструментария. В ходе работ по созданию ЭС сложилась определенная технология их разработки, включающая шесть следующих этапов (рис. 7.5): идентификацию, концептуализацию, формализацию, выполнение, тестирование, опытную эксплуатацию.

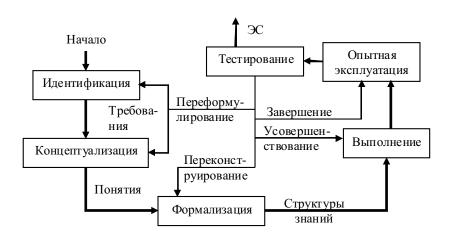


Рис. 7.5 – Технология разработки ЭС.

На этапе идентификации определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей. На этапе концептуализации проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. На этапе формализации выбираются ИС и определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями. На этапе выполнения осуществляется наполнение экспертом базы знаний. В связи с тем, что основой ЭС являются знания, данный этап является наиболее важным и наиболее трудоемким этапом разработки ЭС. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

7.6 Представление знаний в экспертных системах

Первый и основной вопрос, который нужно решить при представлении знаний, — это вопрос определения состава знаний, т. е. определение того, «ЧТО ПРЕД-СТАВЛЯТЬ» в экспертной системе. Второй вопрос касается того, «КАК ПРЕД-СТАВЛЯТЬ» знания. Необходимо отметить, что эти две проблемы не являются независимыми. Действительно, выбранный способ представления может оказаться непригодным в принципе либо неэффективным для выражения некоторых знаний.

По нашему мнению, вопрос «КАК ПРЕДСТАВЛЯТЬ» можно разделить на две в значительной степени независимые задачи: как организовать (структурировать) знания и как представить знания в выбранном формализме.

Стремление выделить организацию знаний в самостоятельную задачу вызвано, в частности, тем, что задача возникает для любого языка представления и способы решения этой задачи являются одинаковыми (либо сходными) вне зависимости от используемого формализма.

Итак, в круг вопросов, решаемых при представлении знаний, будем включать следующие:

- 1) определение состава представляемых знаний;
- 2) организацию знаний;
- 3) представление знаний, т. е. определение модели представления (глава 6). **Состав знаний** ЭС определяется следующими факторами:
- проблемной средой;
- архитектурой экспертной системы;
- потребностями и целями пользователей;
- языком общения.

В соответствии с общей схемой статической экспертной системы (см. рис. 7.4) для ее функционирования требуются следующие знания:

- 1) знания о процессе решения задачи (т. е. управляющие знания), используемые интерпретатором (решателем);
- 2) знания о языке общения и способах организации диалога, используемые лингвистическим процессором (диалоговым компонентом);
- 3) знания о способах представления и модификации знаний, используемые компонентами приобретения и объяснения;
- 4) поддерживающие структурные и управляющие знания, используемые объяснительным компонентом.

Для динамической ЭС, кроме того, необходимы следующие знания:

- 1) знания о методах взаимодействия с внешним окружением;
- 2) знания о модели внешнего мира.

Зависимость состава знаний от требований пользователя проявляется в следующем:

• с какими задачами (из общего набора задач) и с какими данными хочет работать пользователь;

- каковы предпочтительные способы и методы решения;
- при каких ограничениях на количество результатов и способов их получения должна быть решена задача;
- каковы требования к языку общения и организации диалога;
- какова степень общности (конкретности) знаний о проблемной области, доступная пользователю;
- каковы цели пользователей.

Состав знаний о языке общения зависит как от языка общения, так и от требуемого уровня понимания.

С учетом архитектуры экспертной системы знания целесообразно делить на интерпретируемые и неинтерпретируемые. К первому типу относятся те знания, которые способен интерпретировать решатель (интерпретатор). Все остальные знания относятся ко второму типу. Решатель не знает их структуры и содержания. Если эти знания используются каким-либо компонентом системы, то он не «осознает» этих знаний. Неинтерпретируемые знания подразделяются на вспомогательные, хранящие информацию о лексике и грамматике языка общения, информацию о структуре диалога, и поддерживающие знания. Вспомогательные знания обрабатываются естественно — языковой компонентой, но ход этой обработки решатель не осознает, так как этот этап обработки входных сообщений является вспомогательным для проведения экспертизы. Поддерживающие знания используются при создании системы и при выполнении объяснений. Поддерживающие знания выполняют роль описаний (обоснований) как интерпретируемых знаний, так и действий системы. Поддерживающие знания подразделяются на семантические и технологические. Технологические поддерживающие знания содержат знания о времени создания описываемых ими знаний, об авторе знаний и т.п. Семантические поддерживающие знания содержат смысловое описание этих знаний. Они содержат информацию о причинах ввода знаний, о назначении знаний, описывают способ использования знаний и получаемый эффект. Поддерживающие знания имеют описательный характер.



Интерпретируемые знания можно разделить на предметные знания, управляющие знания и знания о представлении.

.....

Знания о представлении содержат информацию о том, каким образом (в каких структурах) в системе представлены интерпретируемые знания.

Предметные знания содержат данные о предметной области и способах преобразования этих данных при решении поставленных задач. Отметим, что по отношению к предметным знаниям знания о представлении и знания об управлении являются метазнаниями. В предметных знаниях можно выделить описатели и собственно предметные знания. Описатели содержат определенную информацию о предметных знаниях, такую, как коэффициент определенности правил и данных, меры важности и сложности. Собственно предметные знания разбиваются на факты и исполняемые утверждения. Факты определяют возможные значения

сущностей и характеристик предметной области. Исполняемые утверждения содержат информацию о том, как можно изменять описание предметной области в ходе решения задач. Говоря другими словами, исполняемые *утверждения*— это знания, задающие процедуры обработки. Однако мы избегаем использовать термин «процедурные знания», так как хотим подчеркнуть, что эти знания могут быть заданы не только в процедурной, но и в декларативной форме.



Управляющие знания можно разделить на фокусирующие и решающие.

Фокусирующие знания содержат сведения о наиболее перспективных объектах или правилах, которые целесообразно использовать при проверке соответствующих гипотез. В первом случае внимание фокусируется на элементах рабочей памяти, во втором — на правилах базы знаний. Решающие знания содержат информацию, используемую для выбора способа интерпретации знаний, подходящего к текущей ситуации. Эти знания применяются для выбора стратегий или эвристик, наиболее эффективных для решения данной задачи.

Качественные и количественные показатели экспертной системы могут быть значительно улучшены за счет использования метазнаний, т. е. знаний о знаниях.

Метазнания не представляют некоторую единую сущность, они могут применяться для достижения различных целей. Перечислим возможные назначения метазнаний:

- 1) метазнания в виде стратегических метаправил используются для выбора релевантных правил;
- 2) метазнания используются для обоснования целесообразности применения правил из области экспертизы;
- 3) метазнания используются для обнаружения синтаксических и семантических ошибок в предметных правилах;
- 4) метаправила позволяют системе адаптироваться к окружению путем перестройки предметных правил и функций;
- 5) метаправила позволяют явно указать возможности и ограничения системы, т. е. определить, что система знает, а что не знает.

Вопросы организации знаний необходимо рассматривать в любом представлении, и их решение в значительной степени не зависит от выбранного способа (модели) представления. Выделим следующие аспекты проблемы организации знаний:

- организация знаний по уровням представления и по уровням детальности;
- организация знаний в рабочей памяти;
- организация знаний в базе данных.

Уровни представления и уровни детальности. Для того чтобы ЭС могла управлять процессом поиска решения, была способна приобретать новые знания и объяснять свои действия, она должна уметь не только использовать свои знания, но и обладать способностью понимать и исследовать их, т. е. ЭС должна иметь

знания о том, как представлены ее знания о проблемной среде. Если знания о проблемной среде назвать знаниями нулевого уровня представления, то первый уровень представления содержит метазнания, т. е. знания о том, как представлены во внутреннем мире системы знания нулевого уровня. Первый уровень содержит знания о том, какие средства используются для представления знаний нулевого уровня. Знания первого уровня играют существенную роль при управлении процессом решения, при приобретении и объяснении действий системы. В связи с тем, что знания первого уровня не содержат ссылок на знания нулевого уровня, знания первого уровня независимы от проблемной среды.

Число уровней представления может быть больше двух. Второй уровень представления содержит сведения о знаниях первого уровня, т. е. знания о представлении базовых понятий первого уровня. Разделение знаний по уровням представления обеспечивает расширение области применимости системы.

Выделение уровней детальности позволяет рассматривать знания с различной степенью подробности. Количество уровней детальности во многом определяется спецификой решаемых задач, объемом знаний и способом их представления. Как правило, выделяется не менее трех уровней детальности, отражающих соответственно общую, логическую и физическую организацию знаний. Введение нескольких уровней детальности обеспечивает дополнительную степень гибкости системы, так как позволяет производить изменения на одном уровне, не затрагивая другие. Изменения на одном уровне детальности могут приводить к дополнительным изменениям на этом же уровне, что оказывается необходимым для обеспечения согласованности структур данных и программ. Однако наличие различных уровней препятствует распространению изменений с одного уровня на другие.

Организация знаний в рабочей системе. Рабочая память (РП) экспертных систем предназначена для хранения данных. Данные в рабочей памяти могут быть однородны или разделяются на уровни по типам данных. В последнем случае на каждом уровне рабочей памяти хранятся данные соответствующего типа. Выделение уровней усложняет структуру экспертной системы, но делает систему более эффективной. Например, можно выделить уровень планов, уровень агенды (упорядоченного списка правил, готовых к выполнению) и уровень данных предметной области (уровень решений).

В современных экспертных системах данные в рабочей памяти рассматриваются как изолированные или как связанные. В первом случае рабочая память состоит из множества простых элементов, а во втором—из одного или нескольких (при нескольких уровнях в РП) сложных элементов (например, объектов). При этом сложный элемент соответствует множеству простых, объединенных в единую сущность. Теоретически оба подхода обеспечивают полноту, но использование изолированных элементов в сложных предметных областях приводит к потере эффективности.



Данные в РП в простейшем случае являются *константами* и (или) *переменными*.

При этом переменные могут трактоваться как характеристики некоторого объекта, а константы — как значения соответствующих характеристик. Если в РП требуется анализировать одновременно несколько различных объектов, описывающих текущую проблемную ситуацию, то необходимо указывать, к каким объектам относятся рассматриваемые характеристики. Одним из способов решения этой задачи является явное указание того, к какому объекту относится характеристика.

Если РП состоит из сложных элементов, то связь между отдельными объектами указывается явно, например заданием семантических отношений. При этом каждый объект может иметь свою внутреннюю структуру. Необходимо отметить, что для ускорения поиска и сопоставления данные в РП могут быть связаны не только логически, но и ассоциативно.

Организация знаний в базе данных. Показателем интеллектуальности системы с точки зрения представления знаний считается способность системы использовать в нужный момент необходимые (релевантные) знания. Системы, не имеющие средств для определения релевантных знаний, неизбежно сталкиваются с проблемой «комбинаторного взрыва». Можно утверждать, что эта проблема является одной из основных причин, ограничивающих сферу применения экспертных систем.



В проблеме доступа к знаниям можно выделить три аспекта: связность знаний и данных, механизм доступа к знаниям и способ сопоставления.

......

Связность (агрегация) знаний является основным способом, обеспечивающим ускорение поиска релевантных знаний. Большинство специалистов пришли к убеждению, что знания следует организовывать вокруг наиболее важных объектов (сущностей) предметной области. Все знания, характеризующие некоторую сущность, связываются и представляются в виде отдельного объекта. При подобной организации знаний, если системе потребовалась информация о некоторой сущности, то она ищет объект, описывающий эту сущность, а затем уже внутри объекта отыскивает информацию о данной сущности. В объектах целесообразно выделять два типа связок между элементами: внешние и внутренние. Внутренние связки объединяют элементы в единый объект и предназначены для выражения структуры объекта. Внешние связки отражают взаимозависимости, существующие между объектами в области экспертизы. Многие исследователи классифицируют внешние связки на логические и ассоциативные. Логические связки выражают семантические отношения между элементами знаний. Ассоциативные связки предназначены для обеспечения взаимосвязей, способствующих ускорению процесса поиска релевантных знаний.

Основной проблемой при работе с большой базой знаний является проблема поиска знаний, релевантных решаемой задаче. В связи с тем, что в обрабатываемых данных может не содержаться явных указаний на значения, требуемые для их обработки, необходим более общий механизм доступа, чем метод прямого доступа (метод явных ссылок). Задача этого механизма состоит в том, чтобы по некоторому описанию сущности, имеющемуся в рабочей памяти, найти в базе знаний объекты,

удовлетворяющие этому описанию. Очевидно, что упорядочение и структурирование знаний могут значительно ускорить процесс поиска.

Нахождение желаемых объектов в общем случае уместно рассматривать как двухэтапный процесс. На первом этапе, соответствующем процессу выбора по ассоциативным связкам, совершается предварительный выбор в базе знаний потенциальных кандидатов на роль желаемых объектов. На втором этапе путем выполнения операции сопоставления потенциальных кандидатов с описаниями кандидатов осуществляется окончательный выбор искомых объектов. При организации подобного механизма доступа возникают определенные трудности: Как выбрать критерий пригодности кандидата? Как организовать работу в конфликтных ситуапиях? и т. п.

Операция сопоставления может использоваться не только как средство выбора нужного объекта из множества кандидатов; она может быть использована для классификации, подтверждения, декомпозиции и коррекции. Для идентификации неизвестного объекта он может быть сопоставлен с некоторыми известными образцами. Это позволит классифицировать неизвестный объект как такой известный образец, при сопоставлении с которым были получены лучшие результаты. При поиске сопоставление используется для подтверждения некоторых кандидатов из множества возможных. Если осуществлять сопоставление некоторого известного объекта с неизвестным описанием, то в случае успешного сопоставления будет осуществлена частичная декомпозиция описания.

Операции сопоставления весьма разнообразны. Обычно выделяют следующие их формы: синтаксическое, параметрическое, семантическое и принуждаемое сопоставления. В случае синтаксического сопоставления соотносят формы (образцы), а не содержание объектов. Успешным является сопоставление, в результате которого образцы оказываются идентичными. Обычно считается, что переменная одного образца может быть идентична любой константе (или выражению) другого образца. Иногда на переменные, входящие в образец, накладывают требования, определяющие тип констант, с которыми они могут сопоставляться. Результат синтаксического сопоставления является бинарным: образцы сопоставляются или не сопоставляются. В параметрическом сопоставлении вводится параметр, определяющий степень сопоставления. В случае семантического сопоставления соотносятся не образцы объектов, а их функции. В случае принуждаемого сопоставления один сопоставляемый образец рассматривается с точки зрения другого. В отличие от других типов сопоставления здесь всегда может быть получен положительный результат. Вопрос состоит в силе принуждения. Принуждение могут выполнять специальные процедуры, связываемые с объектами. Если эти процедуры не в состоянии осуществить сопоставление, то система сообщает, что успех может быть достигнут только в том случае, если определенные части рассматриваемых сущностей можно считать сопоставляющимися.

Методы поиска решений в экспертных системах. Методы решения задач, основанные на сведении их к поиску, зависят от психодиагностики в психосоматике, от особенностей предметной области, в которой решается задача, и от требований, предъявляемых пользователем к решению. Особенности предметной области с точки зрения методов решения можно характеризовать следующими параметрами:

- размер, определяющий объем пространства, в котором предстоит искать решение;
- изменяемость области, характеризует степень изменяемости области во времени и пространстве (здесь будем выделять статические и динамические области);
- полнота модели, описывающей область, характеризует адекватность модели, используемой для описания данной области. Обычно если модель не полна, то для описания области используют несколько моделей, дополняющих друг друга за счет отражения различных свойств предметной области;
- определенность данных о решаемой задаче, характеризует степень точности (ошибочности) и полноты (неполноты) данных. Точность (ошибочность) является показателем того, что предметная область с точки зрения решаемых задач описана точными или неточными данными; под полнотой (неполнотой) данных понимается достаточность (недостаточность) входных данных для однозначного решения задачи. Требования пользователя к результату задачи, решаемой с помощью поиска, можно характеризовать количеством решений и свойствами результата и (или) способом его получения. Параметр «количество решений» может принимать следующие основные значения: одно решение, несколько решений, все решения. Параметр «свойства» задает ограничения, которым должен удовлетворять полученный результат или способ его получения. Так, например, для системы, выдающей рекомендации по лечению больных, пользователь может указать требование не использовать некоторое лекарство (в связи с его отсутствием или в связи с тем, что оно противопоказано данному пациенту). Параметр «свойства» может определять и такие особенности, как время решения («не более чем», «диапазон времени» и т.п.), объем памяти, используемой для получения результата, указание об обязательности (невозможности) использования каких-либо знаний (данных) и т. п.

Итак, сложность задачи, определяемая вышеприведенным набором параметров, варьируется от простых задач малой размерности с неизменяемыми определенными данными и отсутствием ограничений на результат и способ его получения до сложных задач большой размерности с изменяемыми, ошибочными и неполными данными и произвольными ограничениями на результат и способ его получения. Из общих соображений ясно, что каким-либо одним методом нельзя решить все задачи. Обычно одни методы превосходят другие только по некоторым из перечисленных параметров. Рассмотренные ниже методы могут работать в статических и динамических проблемных средах. Для того чтобы они работали в условиях динамики, необходимо учитывать время жизни значений переменных, источник данных для переменных, а также обеспечивать возможность хранения истории значений переменных, моделирования внешнего окружения и оперирования временными категориями в правилах.

Существующие методы решения задач, используемые в экспертных системах, можно классифицировать следующим образом:

1) методы поиска в одном пространстве — методы, предназначенные для использования в следующих условиях: области небольшой размерности, полнота модели, точные и полные данные;

- 2) методы поиска в иерархических пространствах методы, предназначенные для работы в областях большой размерности;
- 3) методы поиска при неточных и неполных данных;
- 4) методы поиска, использующие несколько моделей, предназначенные для работы с областями, для адекватного описания которых одной модели недостаточно.

Предполагается, что перечисленные методы при необходимости должны объединяться для того, чтобы позволить решать задачи, сложность которых возрастает одновременно по нескольким параметрам.

7.7 Блок (подсистема) объяснений

Определение блока объяснений. Каждая ЭС имеет подсистему объяснений, которая позволяет, если необходимо, разъяснить пользователю, как ЭС получила то или иное решение. При этом широко используются сведения из базы знаний, с которыми согласен пользователь. Именно они являются аргументами, которые приводит ЭС, объясняя свое решение. Наличие такого объяснения делает полученное решение понятным и «прозрачным» пользователю. (Известно, что человек плохо воспринимает, а иногда и просто отвергает необоснованные советы. В этом проявляется критицизм—важная черта, защищающая его от слепого подчинения, свойственного, например, роботам, да и то не всем) [12].

П. Джексон [4] отмечает, что представление информации о поведении ЭС важно по многим причинам.

- 1) *Пользователи*, работающие с системой, нуждаются в подтверждении того, что в каждом конкретном случае заключение, к которому пришла программа, в основном корректно.
- 2) Инженеры, имеющие дело с формированием базы знаний, должны убедиться, что сформулированные ими знания применены правильно, в том числе в случае, когда существует прототип.
- 3) Экспертам в предметной области желательно проследить ход рассуждений и способ использования тех сведений, которые с их слов были введены в базу знаний. Это позволит судить, насколько корректно они применяются в данной ситуации.
- 4) *Программистам*, которые сопровождают, отлаживают и модернизируют систему, нужно иметь в своем распоряжении инструмент, позволяющий заглянуть в «ее нутро» на уровне более высоком, чем вызов отдельных языковых процедур.
- 5) *Менеджер системы*, использующей экспертную технологию, который в конце концов несет ответственность за последствия решения, принятого программой, также нуждается в подтверждении, что эти решения достаточно обоснованы.

Назначение блока объяснений. Блок объяснений на вопрос «Как получить именно это решение?» дает все необходимые объяснения, используя информацию, которая в процессе выработки решения накапливается в логическом блоке. Как

правило, сценарий предусматривает несколько путей; среди них возможны тупиковые пути, не ведущие к нужной цели. Оказавшись на дороге в тупик, система может автоматически вернуться к той «развилке», где она сделала неверный выбор. Неудачные попытки в памяти логического блока не сохраняются, и в результате к концу поиска решения в нем запоминается лишь результативный путь — трек, ведущий от начальной ситуации к решению. Этот трек и используется блоком объяснения для ответа на запрос пользователя.

Но пользователь может задать и иной вопрос: «Почему рекомендуется решение P1, а не решение P2?». Чтобы ответить на этот вопрос, система должна построить путь от P2 к исходной ситуации, т. е. от готового решения прийти к исходным данным. Далее возможны два варианта: либо система убеждается, что P2 не может быть получено из исходных данных, либо она получает новый трек, который от исходных данных ведет не к P1, а к P2. Сравнивая трек и для P1, и для P2, система находит тот момент, когда она предпочла двигаться к P1, а не к P2, и свои соображения по выбору на этом шаге она сообщает пользователю.

Наконец, пользователь может слабо разбираться в той предметной области, для работы с которой предназначена экспертная система. Например, он может не знать значения некоторых терминов, которые в общении с ним использует система. В этом случае пользователь может задать блоку объяснений вопрос: «Что такое X?» Для ответа на такой вопрос система обращается к энциклопедической информации, хранящейся в базе знаний, — логический блок находит соответствующую статью, где разъясняется, что такое X, и выдает ее текст пользователю.

Три типа вопросов (как, почему, что), — это сегодняшний уровень работы блоков объяснений экспертных систем. В ближайшем будущем эти блоки смогут отвечать и на иные вопросы (зачем, какой и т. п.). Объяснительная возможность экспертных систем в этом случае возрастет. Соответственно возрастет и доверие пользователя к тем решениям, которые ему рекомендует система.

Необходимость использования блока объяснений. П. Джексон [4] выделяет две причины, которые побуждают разработчиков экспертных систем делать их по возможности «прозрачными» для пользователя. Под прозрачностью при этом понимается способность системы объяснить пользователю, почему принято именно такое решение, вследствие каких рассуждений система пришла к тому или иному выводу.

Клиент, который обращается к экспертной системе за советом, должен знать, на основе каких логических доводов этот совет был сформирован. Только получив исчерпывающую информацию о ходе рассуждений, клиент может с доверием отнестись к полученному совету, поскольку за последствия неверно принятого решения расплачиваться придется не столько «советчику», сколько чересчур доверчивому клиенту.

Инженер, обслуживающий экспертную систему, должен быть уверен в правильности работы всех подсистем, а проверить это он может, только получив от экспертной системы всю возможную информацию о ходе рассуждений в процессе решения задач.

Развитие подсистемы объяснений. На начальном этапе исследований в области экспертных систем, которые выполнялись в Станфордском университете в 60–70-х годах, поясняющая информация предоставлялась в виде трассировки про-

цесса выполнения программы и использовалась в основном для отладки разрабатываемых систем. Этого было достаточно для разработчиков экспериментальных систем, но не соответствовало тому уровню сервиса пользователя, который необходим для коммерческого программного продукта. Впоследствии вопросу формирования информации, которая давала бы возможность пользователю четко представить себе ход рассуждений программы, стало уделяться значительно больше внимания.



Исследователи пришли к заключению, что автоматическое формирование пояснений требует доступа к *модели предметной области* точно так же, как и извлечение знаний.

Другими словами, представление о знаниях в конкретной области необходимо для предоставления пользователю информации о поведении системы в процессе формирования результата точно так же, как и для приобретения новых знаний.

Такое знание позволит перекинуть мост между деталями реализации процесса вывода (например, в какой последовательности активизировались правила) и стратегией поведения системы (например, какие соображения побудили систему выбрать ту или иную гипотезу из множества конкурирующих).

В последние десятилетия специалисты серьезно потрудились над развитием этой идеи. Совершенно очевидно, что проблемы извлечения знаний и формирования пояснений тесно связаны. По сути, они представляют две стороны одной медали. Существенным толчком для совершенствования средств, используемых для предоставления пользователю пояснений, как, впрочем, и для извлечения знаний, стало развитие методов графического интерфейса в современных операционных системах, которые обеспечивают возможность вывода не только статической, но и динамической видеоинформации со звуковым сопровождением.



Пример функционирования подсистемы объяснений. Ниже излагается один из известных подходов к созданию диагностирующих продукционных ЭС, приведенный в книге И. Г. Черноруцкого [6]. Соответствующие интерпретации могут быть весьма разнообразны, и поэтому область применимости обсуждаемых конструкций оказывается достаточно широкой.

Задано множество фактов $A = \{a_{ij}\} \cup \{q_i\} = (a_1, a_2, \ldots, a_n)$, состоящее из элементов двух типов. Элементы a_{ij} определяют обычные декларативные знания из конкретной предметной области. Элементы q_i определяют вид взаимодействия с внешней средой и в данном случае представляют собой вопросы пользователю в виде альтернативного меню $q_i = a_{i1}, \ldots, a_{in}$.

Некоторые из q_i имеют другой смысл — результирующих значений или диагнозов, оформленных в виде соответствующих сообщений пользователю.

Продукции в данной системе имеют вид $a_{ij} \to q_m = \{a_{m1}, \dots, a_{mk}\}.$

Все множество фактов и продукций организовано в некоторую систему,

представленную в виде графа «ИЛИ». Фрагменты такого диагностирующего графа с вершинами-диагнозами $q_9, q_{10}, q_{11}, q_{12}$ (терминальными вершинами) представлен на рис. 7.6.

Принцип работы такой ЭС заключается в следующем. После обращения пользователя к ЭС мы попадаем в вершину q_1 , инициирующую вопрос пользователю в виде соответствующего альтернативного меню: $q_1 = \{a_{12}, a_{13}, \dots\}$.

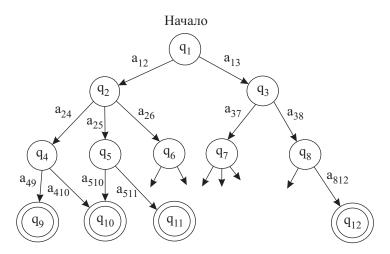


Рис. 7.6 – Диагностирующий граф.

Допустим, на вопрос системы «Какой из фактов $a_{12}, a_{13} \dots$ имеет место?» пользователь ответил: a_{12} . В результате мы заносим a_{12} в рабочее поле и попадаем в новую вершину q_2 : $q_2 = \{a_{24}, a_{25}, a_{26}\}$, где ситуация повторяется. В конце концов, мы оказываемся в одной из терминальных вершин, где пользователь получает сообщение о результате, характеризующим выбранный ЭС диагноз.

При такой структуре ЭС достаточно просто может быть реализована подсистема объяснений как важнейшая составная часть любой ЭС. Для этого достаточно каждой вершине q_i графа сопоставить соответствующий текст, описывающий мотивации выбора в данной вершине. А далее происходит вывод этих текстов при движении снизу (от результата) вверх в соответствии с реализованной цепочкой рассуждений. Для ответа на вопросы пользователей типа «Почему q_i , а не q_j ?» система, двигаясь по цепочке снизу вверх, определяет место разветвления (какуюто вершину q_k) путей, ведущих к q_i , и q_j . Пояснительный текст, ассоциированный с вершиной q_k , и должен содержать ответ на вопрос пользователя. Например, на вопрос «Почему q_9 , а не q_{12} ?» система ответит «Потому что a_{12} , а не a_{13} », а на вопрос «Почему q_9 , а не q_{11} » система ответит «Потому что a_{24} , а не a_{25} » и т. п.



Пример общения с подсистемой объяснений. В своей книге Ж.-Л Лорьер [9] приводит следующий пример с экспертной системой MYCIN и системой приобретения знаний TEIRESIAS.

MYCIN: Произошло ли инфицирование больного в больнице?

Пользователь: Почему?

TEIRESIAS: (Почему это важно знать?) Это поможет определить, к какой категории относится микроорганизм, ответственный за инфекцию.

- Уже установлено, что, во-первых, организм грамотрицателен; во-вторых, это бактерия и, в-третьих, она аэробна или анаэробна.
- Следовательно, заражение произошло не во время пребывания в больнице.
- Почти наверное (0.8), что организм относится к категории энтеробактерий; более того, возможно (0.1 правило 37), что это pseudo-monia aeruginosa.

Пользователь: Почему вы не предположили возможности того, что причиной заболевания является стрептококк?

TEIRESIAS: Для того чтобы идентифицировать организм как стрептококк, нужно было использовать правило 33, но уже известна сложность его второй предпосылки, а именно «морфологический тип — оболочка», поэтому правило 33 неприменимо.

7.8 Взаимодействие пользователя с экспертной системой

Взаимодействие с ЭС осуществляют различные типы пользователей: специалисты, неспециалисты, учащиеся. Каждый из них имеет определенные требования к диалогу, но всех их объединяет следующее: языком общения должен быть ограниченный естественный язык (ЕЯ) или близкий к нему; процесс взаимодействия пользователей с ЭС не сводится к изолированным парам «запрос-ответ», а представляет собой разветвленный диалог, в котором инициатива переходит от одного участника к другому [1, 3].

7.8.1 Структура системы с естественным языком общения

Основные понятия.



Диалог — процесс достижения его участниками определенных согласованных целей путем обмена связанными высказываниями, выраженными в языке, о некотором реальном или гипотетическом мире (проблемной области).

.....

......

Говоря об общении (*Общение* — коммуникативное взаимодействие) человека с компьютером, обычно считают, что цель компьютера состоит в том, чтобы способствовать достижению целей пользователя, которые определяются его информационными потребностями. Поэтому применительно к диалогу между пользователем и компьютером под общением понимают процесс обмена взаимосвязанными

высказываниями, выраженными в языке, направленный на достижение целей пользователя, т. е. на удовлетворение информационных потребностей пользователя (ИПП).

В общем случае процесс общения не может быть сведен к обмену изолированными парами высказываний «вопрос-ответ». Высказывания участников общения образуют связный текст — дискурс, имеющий, как правило, достаточно сложную структуру. Связность дискурса обеспечивается как лингвистическими (родовидовыми, анафорическими, модальными, стилистическими согласованиями, согласованиями пресуппозиций и т.п.), так и экстралингвистическими (ситуативными) средствами, т. е. с помощью временных, причинно-следственных и других связей, существующих в проблемной области.

Цели, преследуемые участниками общения, определяют структуру диалога, которая может рассматриваться на трех уровнях:

- глобальном;
- тематическом;
- локальном.

На глобальном уровне определяются общие свойства решаемых пользователями задач.

На тематическом уровне структура диалога зависит от конкретных особенностей решаемой задачи — от алгоритма ее решения (разбиения задач на подзадачи) и распределения ролей (активная или пассивная роль) между участниками общения при решении отдельных подзадач. На локальном уровне рассматриваются отдельные шаги диалога, образуемые взаимосвязанными высказываниями его участников.

Шаг диалога трактуется как пара «действие-реакция», где высказывание активного (т. е. владеющего инициативой) участника соответствует действию, а пассивного — реакции.

Основными параметрами структуры диалога на этом уровне являются:

- инициатор шага и вид инициирования (вид действия);
- способ влияния действия на реакцию;
- способ спецификации задачи (подзадачи), решаемой на данном шаге.

Действие и реакция, образующие шаг диалога, могут в общем случае не соответствовать соседним (во временной последовательности) высказываниям участников.

Соответствие нарушается при перехватах инициативы. Перехват инициативы возникает в тех случаях, когда пассивный участок вместо цели (подцели), предложенной активным участником, выбирает иные цели (подцели), в частности подцели, предусматривающие преодоление локальных неудач. Например, вместо ответа на вопрос (что соответствовало бы стандартной реакции) второй участник может задать встречный вопрос (т. е. совершить действие и тем самым взять на себя активную роль) и, лишь после получения ответа на него, ответить на первоначально заданный вопрос (и тем самым вернуть инициативу). Таким образом, перехват инициативы как бы разрывает первоначально инициативный шаг диалога и открывает поддиалог — происходит смена цели (темы) диалога, в котором инициативой владеет ранее пассивный участник.

Традиционные средства общения, которые вплоть до настоящего времени широко применяются на практике, ориентированы, как правило, только на использование, разработку и развитие. В первом случае процесс взаимодействия сводится к трем этапам:

- определение параметров работы системы (вход в систему);
- определение решаемой задачи и исходных данных;
- получение результатов решения задачи.

Такой процесс принципиально не может удовлетворять пользователей с изменяющейся информационной потребностью, не знающих способа представления и использования в системе информации, которой обмениваются участники общения.

Во втором случае взаимодействие осуществляется с помощью процедурного языка программирования, что не удовлетворяет большинство конечных пользователей, обычно не умеющих (и не желающих) программировать.

Низкая эффективность, а часто неприемлемость традиционных средств общения в большинстве случаев вызвана тем, что в них не учитываются важнейшие особенности процесса общения. Эти особенности, независимо от специфики решаемых пользователями задач, сводятся к следующим:

- 1) Изменяемость. Информационная потребность пользователя не может быть заранее четко определена в спецификациях на разработку системы общения, напротив, ИПП неизбежно изменяется в ходе разработки и эксплуатации системы.
- 2) Несовпадение взглядов на мир. Представления, имеющиеся у пользователя и системы о языке общения и проблемной области, относительно которой ведется общение, могут не совпадать. Исходя из этого процесс общения должен предусматривать разъяснение смысла неизвестных терминов, обнаружение и устранение несовпадающих представлений, а также предупреждение ошибочных толкований, т. е. установление общих точек зрения на обсуждаемые в процессе общения сущности.
- 3) Связность общения. Процесс общения не может быть ограничен обменом изолированными парами «вопрос-ответ», так как в большинстве реальных случаев ИПП не может быть выражена в виде одного вопроса (предложения). Часто требуется определить ситуацию, в которой возникла ИПП, т. е. предпослать запросу на решение некоторой задачи контекст, в котором эту задачу необходимо решать. Кроме того, процесс удовлетворения ИПП решение некоторой задачи в большинстве реальных приложений требует взаимодействия, основанного, на смешанной инициативе участников. Поэтому процесс общения должен иметь сложную, разветвленную структуру и состоять из обмена связанными высказываниями.
- 4) «Неправильность» высказываний пользователя. Для выражения ИПП пользователь может применить как «правильные» предложения, т. е. такие, которые будут однозначно поняты и верно обработаны системой, так и «неправильные». Неправильности могут быть вызваны, во-первых, тем, что пользователь обычно не в состоянии учесть все ограничения системы общения в части ее возможностей и знаний, во-вторых, использованием умолчаний,

характерных для естественного общения и допускающих неоднозначное толкование высказываний, и, в-третьих, отклонением предложений от грамматической нормы.

Недостатки традиционных средств общения потребовали создания средств нового поколения, которые должны быть способны настраиваться на ИПП и адаптироваться к их изменению, представлять и объяснять свою точку зрения на проблемную область, а также учитывать точку зрения пользователя, поддерживать связный диалог и уметь обрабатывать «неправильные» высказывания. Разработка этих средств ведется в настоящее время по двум основным направлениям. Первое направление, развиваемое преимущественно специалистами по системам обработки данных, заключается в повышении уровня и увеличении непроцедурности формализованных языков общения. Типичными представителями таких языков являются, например, APL, NOMAD, MAPPER. Второе направление развивается в рамках искусственного интеллекта и предполагает использование конечными пользователями для взаимодействия с компьютером естественного языка, семантически и прагматически ограниченного проблемной областью, относительно которой ведется общение.

Традиционные средства общения не позволяют обеспечить взаимодействие конечных пользователей с компьютером. Чтобы быть полноправным участником общения, ЕЯ-система должна выполнять некоторые обязательные функции. К этим функциям относятся следующие.

Ведение диалога — определение его структуры и ранга роли, которую система и пользователь выполняют на текущем шаге диалога.

Понимание — преобразование поступающих от пользователя высказываний на естественном языке в высказывания на языке внутреннего представления.

Обработка высказываний — формирование или определение заданий на решение задач или подзадач на данном шаге диалога.

Генерация — формирование выходных высказываний на ЕЯ.

Приведенные функции имеют обобщенный характер. Поэтому необходимо подчеркнуть, что при реализации конкретных ЕЯ-систем суть этих функций может в значительной степени варьироваться. В соответствии с выделенными функциями обобщенная схема ЕЯ-системы (рис. 7.7) может быть представлена в виде трех компонентов: диалоговый; компонент понимания высказываний; компонент генерации высказываний.

Диалоговый компонент. К основным задачам диалогового компонента относятся:

- ведение диалога;
- формирование или обработка перехватов инициативы.

Ведение диалога состоит в том, чтобы обеспечивать целесообразные (т. е. способствующие достижению конечных целей пользователя) действия системы на текущем шаге диалога. В связи с тем, что возможности существующих ЕЯ-систем не позволяют им самостоятельно формировать целесообразное поведение, в систему обычно вводится информация, определяющая общую и тематическую структуры диалога. По структуре и текущему состоянию диалога диалоговый компонент формирует (если инициатива принадлежит системе) или определяет (если инициатива принадлежит пользователю) задание, выполняемое системой на текущем шаге (на-

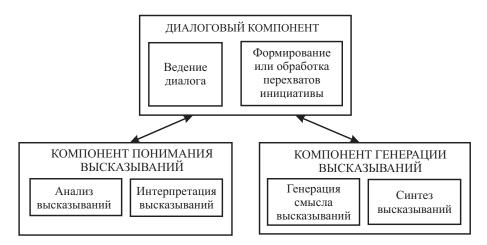


Рис. 7.7 – Обобщенная схема Е-Я-системы.

пример, генерация вопроса, понимание ответа и его обработка, генерация утверждения и т. п.).

Ведение диалога выполняется по одной из двух схем:

- диалог ведет пользователь;
- диалог ведет система.

В первом случае инициатива в основном (за исключением сообщений об ошибках) находится у пользователя, а система только реагирует на его требования, определяя по виду требования тип задания. Разбиение задачи на подзадачи и распределение ролей осуществляет пользователь, т. е. для системы весь диалог сводится к выработке реакции на текущие высказывания пользователя.

Можно сказать, что в этом варианте работы функции диалогового компонента практически вырождены. Во втором случае инициатива в основном принадлежит системе. Система ведет диалог в соответствии с имеющимися у нее представлениями о структуре диалога (т. е. о разбиении задач на подзадачи и о том, кто из участников, когда и какую подзадачу решает) и о способе обмена высказываниями. Если роли участников неизменны, однозначны и предопределены заранее, то структуру диалога называют жесткой. В простейшем случае такая структура диалога сводится к двум взаимосвязанным высказываниям участников (вопрос-ответ) с указанием участника, владеющего инициативой. Развитием жесткой структуры является альтернативная структура, которая задает множество возможных (но заранее предписанных) направлений течения диалога. Выбор одного из возможных направлений осуществляет пассивный участник. Если роли участников общения распределяются в ходе общения, то структуру диалога называют гибкой. Гибкие структуры подразделяются по степени свободы выбора момента перехвата (предопределенные моменты, произвольные моменты) и по способу перехвата (предопределенный способ перехвата, произвольный способ).

Вторая задача диалогового компонента вызвана тем, что реакции одного участника могут не соответствовать ожиданиям другого. В зависимости от того, кто осуществляет перехват инициативы, система либо формирует перехват, либо обрабатывает его. Формирование происходит в тех случаях, когда система определяет, что текущая ситуация не соответствует ситуации, предусмотренной структурой диалога.

Если же перехват инициативы осуществляет пользователь, то задача системы — обработать его, т. е. распознать наличие перехвата инициативы, определить новую тему (цель), на которую перешел пользователь, и перейти на структуру диалога, соответствующую новой теме.

7.8.2 Компьютерно-лингвистический подход к диалогу

В современной научной литературе наметились два подхода к анализу и построению вопросов и ответов как основных элементов в структуре диалоговой формы передачи информации: лингвистический и компьютерный. Но многообразие видов и областей использования диалога настоятельно требует разработок синтетического характера, в которых достижения лингвистического и компьютерного анализа вопросно-ответных структур были бы представлены в единстве. Благодатной почвой, на которой может быть взращено синтетическое направление в исследовании диалога, оказывается логика. При синтетическом подходе к диалогу на базе логики удается, с одной стороны, формализовать естественно-языковые средства общения и повысить тем самым степень детерминированности основных структурных элементов диалога, а, с другой стороны, максимально приблизить форму компьютерного представления диалога к естественноречевому общению, увеличив тем самым эффективность восприятия информации в человекомашинных системах, К особо важным логическим закономерностям диалога как вида человеческих рассуждений или способа информационного обмена относятся те структурно-информационные характеристики, которые инвариантны к содержанию конкретного диалога.

Лингвистический подход к диалогу. Понятие «естественный язык» знакомо любому человеку. Но в лингвистике нет бесспорного определения этому понятию. Естественный язык—это некоторая система элементов плюс система правил их функционирования.

В русском языке для письменного текста выделяется следующая система элементов: морфемы, слова, словосочетания, предложения, связанный текст (дискурс).

Рассмотренные единицы языка образуют некоторую систему разных уровней. Каждому уровню в языке соответствует группа правил. Исторически выделяются следующие основные группы правил: морфологические, синтаксические, семантические.

Морфологические правила определяют законы сочетания морфем в слове; синтаксические—законы сочетания слов или словосочетаний во фразах и предложениях; семантические—законы преобразования предложения в тексте. Выделение правил и их формализация необходимы для построения моделей ЕЯ.

Естественному языку присущи омонимия и синонимия. Омонимия позволяет использовать ограниченное число знаков для обозначения разных реальных объектов и понятий. Синонимия — характеризовать некоторую сущность различными знаками, подчеркивая определенные ее стороны.

В языке, а особенно в речи, допускаются и часто используются неполные конструкции. Одна из видов таких конструкций — эллипсис, представляющий собой предложение, в котором опущено слово или несколько слов, легко восстанавливаемых по локальному или глобальному дискурсу.

Средства отождествления и различения объектов, упоминаемых в тексте, реализуются с помощью анафор. Анафора обозначает (чаще всего) двукратное повторение в разных частях предложения или в нескольких связанных фразах словесного отображения одного и того же объекта. В частном случае анафора — это повторение одного и того же слова. Одно из таких повторений называется антецедентом, а другое — ссылкой. Чаще всего используются местоименные ссылки.

Известно, что язык задает норму. Однако она не является обязательной для речи. Типичны, например, орфографические ошибки и др. При машинном анализе затруднения в понимании могут вызвать и такие предложения, которые, казалось бы, построены правильно. Естественный язык, кроме того, изобилует идиоматическими и фразеологическими оборотами, метафорами и ассоциациями, нечеткостями и неопределенностями и т. п. Все это затрудняет моделирование ЕЯ.

Однако выход из этой ситуации существует и состоит в том, что при построении систем общения используются подмножества ЕЯ, которые в конкретных приложениях проблемно ориентированы и, следовательно, более просты, чем ЕЯ в полном объеме. Такие подмножества называют языками деловой прозы. Их лексика ограничена предметной областью, смыслы слов более однозначны, отсутствуют метафоры, ассоциации и т. д. Но тем не менее в них используются те же конструкции предложения, эллипсисы и анафоры и т. д. Поэтому проблема анализа текстов даже для этих языков нетривиальна.

Информационный подход к диалогу. В структуре диалога можно выделить вопрос и ответ. Вопрос — это форма мышления, в которой при наличии определенной информации, выраженной в предпосылках вопроса, требуется дать дополнительную информацию, выражаемую в ответе. В обобщенном виде под вопросом понимается некоторое неполное знание, дополнение к которому (до полного) содержится в предполагаемом истинном ответе, то есть вопрос — это часть информации ответа. Иначе говоря, вопрос — это неполная информация, в которой формулируется требование о ее дополнении до полного и однозначно правильного ответа. Вопросы делятся на корректные (заложены истинные предпосылки или на которые можно дать истинный или ложный ответ) и некорректные. Поскольку вопрос может быть задан относительно любого элемента предложения, то известную часть будем называть темой, а неизвестную, которая должна прозвучать в ответе, ремой.

Элементарный акт (шаг) диалога — это пара, состоящая из вопроса и соответствующего ему ответа. В общем случае процесс общения не может быть сведен к обмену изолированными парами высказываний вопрос-ответ. Высказывания участников общения образуют связный текст — дискурс (цепочку элементарных его актов), имеющий, как правило, достаточно сложную структуру.

.....



Контрольные вопросы и задания к главе 7

- 1) Перечислите основные характеристики экспертных систем первого поколения.
- 2) Перечислите основные характеристики экспертных систем второго поколения.
- 3) Опишите отличие характеристик экспертных систем первого и второго поколений.
- 4) Расскажите о характеристиках перспективных экспертных систем.
- 5) Сформулируйте концепцию разработки традиционной экспертной системы.
- 6) Какие недостатки имели экспертные системы первого поколения?
- 7) Представьте особенности архитектур экспертных систем второго поколения.
- 8) Что позволяют средства самостоятельного извлечения знаний из данных, поступающих в экспертную систему в ходе ее создания и эксплуатации?
- 9) Как решается проблема надежности в экспертной системе управления?
- 10) Назовите два основных процесса в искусственном интеллекте.
- 11) Охарактеризуйте интеграционный процесс в искусственном интеллекте.
- 12) Охарактеризуйте разделительный процесс в искусственном интеллекте.
- 13) Представьте классификацию интегрированных экспертных систем.
- 14) Поясните поверхностную интеграцию экспертных систем.
- 15) Опишите интегрированные экспертные системы с глубинной интеграцией компонентов.
- 16) Перечислите классы интегрированных систем искусственного интеллекта.
- 17) Дайте характеристику способов интеграции экспертных систем.
- 18) Что Вы понимаете под гибридизацией экспертных систем?
- 19) Рассмотрите взаимосвязь способов интеграции компонентов экспертной системы с необходимостью гибридизации.
- 20) Составьте и поясните структурную схему экспертной системы.
- 21) Из каких модулей состоит ядро экспертной системы?
- 22) Представители, каких специальностей участвуют в разработке экспертных систем?
- 23) Охарактеризуйте два режима работы экспертной системы.
- 24) Сформулируйте требования, которые необходимы для разработки экспертной системы.
- 25) Приведите технологию разработки экспертной системы, выделив основные этапы.
- 26) Поясните этапы разработки экспертных систем: идентификации, концептуализации, формализации, выполнения.

- 27) Представьте круг вопросов, решаемых при представлении знаний.
- 28) Какие знания необходимы для разработки статических и динамических экспертных систем?
- 29) Опишите интерпретируемые и неинтерпретируемые знания.
- 30) Проанализируйте уровни представления и детальности знаний проблемной среды в экспертной системе.
- 31) Охарактеризуйте организацию знаний в рабочей памяти и в базе знаний экспертной системы.
- 32) Перечислите общие методы решения задач, используемых в экспертных системах.
- 33) Назовите подсистему экспертной системы, которая позволяет разъяснить пользователю, как получено данное решение.
- 34) Кому и по каким причинам необходимо знать о поведении экспертной системы?
- 35) Сформулируйте назначение блока объяснений.
- 36) Проанализируйте диагностирующий граф продукционных экспертных систем.

......

.....



Литература к главе 7

- [1] Андрейчиков А. В. Интеллектуальные информационные системы : учебник / А. В. Андрейчиков, О. Н. Андрейчикова. М: Финансы и статистика, 2006.-424 с.
- [2] Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин [и др.]; под ред. В. Н. Вагина и Д. А. Поспелова. Физматлит, 2004. 704 с.
- [3] Гаскаров Д. В. Интеллектуальные информационные системы: учебник для вузов / Д. В. Гаскаров. М.: Высшая школа, 2003. 431 с.
- [4] Джексон П. Введение в экспертные системы : пер. с англ. : учеб. пособие / П. Джексон. М. : Издательский дом «Вильямс», 2007.
- [5] Загоруйко Н. Г. Прикладные методы анализа данных и знаний / Н. Г. Загоруйко. Новосибирск: ИМ, 1999.
- [6] Искусственный интеллект: в 3 кн. / под ред. Д. А. Поспелова. М. : Радио и связь, 1990. Кн. 2: Модели и методы.

- [7] Искусственный интеллект. Применение в интегрированных производственных системах / под ред. Э. Кьюсиака. М.: Машиностроение, 1991.
- [8] Колесников А. В. Проблемно-структурная технология разработки приложений гибридных интеллектуальных систем // Тр. VII Национальной конференции по искусственному интеллекту с международным участием $KUU-2000.-T.\ 2.-M.: Физматлит,\ 2000.$
- [9] Лорьер Ж-Л. Системы искусственного интеллекта / Ж-Л. Лорьер. М. : Мир, 1991-568 с.
- [10] Нариньяни А. С. Модели или алгоритмы: новая парадигма информационной технологии // Информационные технологии. 1997. №4.
- [11] Статические и динамические экспертные системы / Э. В. Попов [и др.]. М.: Финансы и статистика, 1996.
- [12] Растригин Л. А. С компьютером наедине / Л. А. Растригин. М. : Радио и связь, 1990.-224 с.
- [13] Рыбина Г. В. Задачно-ориентированная методология автоматизированного построения интегрированных экспертных систем для статических проблемных областей // Известия РАН ТиСУ. 1997. №5.
- [14] Рыбина Г. В. Современные экспертные системы: тенденции к интеграции и гибридизации // Приборы и системы. Управление, контроль, диагностика. -2001.-N28.
- [15] Рыбина Г. В. Проектирование систем, основанных на знаниях / Г. В. Рыбина. М. : МИФИ, 1997.
- [16] Чекинов С. Г. Экспертные системы в системах управления: состояние и перспективы // Информационные технологии. 2001. №4. С. 32–36.
- [17] Хаджаев Г. А. Интеллектуальное управление организационными системами / Г. А. Хаджаев. М.: МГТУ, 1997.

Глава 8

ЗНАНИЯ И ИХ ПРЕДСТАВЛЕНИЕ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

8.1 Понятие знания, представление знаний

В настоящее время в исследованиях по искусственному интеллекту (ИИ) выделились 6 направлений:

- Представление знаний.
- Манипулирование знаниями.
- Общение.
- Восприятие.
- Обучение.
- Поведение.

В рамках направления «Представление знаний» решаются задачи, связанные с формализацией и представлением знаний интеллектуальной системы (ИС). Для этого разрабатываются специальные модели представления знаний и языки для описания знаний, выделяются различные типы знаний. Изучаются источники, из которых ИС может черпать знания, и создаются процедуры и приемы, с помощью которых возможно приобретение знаний для ИС. Проблема представления знаний для ИС чрезвычайно актуальна, т. к. ИС — это система, функционирование которой опирается на знания о проблемной области, которые хранятся в ее памяти.

Прежде, чем перейти к модели представления знаний, определим понятие знание, а затем и отличие его от данных. Пока строгого понятия знания не существует, оно менялось, пополнялось еще с древних греков (Платон и Аристотель). Приведем наиболее распространенные в отечественной литературе понятия этого термина.

1) В Большой Энциклопедии (под ред. С. Н. Южакова, Спб, т. 9, 1902 г.):

«Знание в объективном смысле то же, что истинное знание (познание), в субъективном — убеждение в его истине по реальным причинам. В первом отношении оно противополагается заблуждению, как неистинному знанию, во втором — вере, мнению».

- 2) Знание это обоснованное, истинное убеждение (Вера).
- 3) Из толкового словаря С. И. Ожегова: 1) «Знание постижение действительности сознанием, наука», 2) «Знание это результат, полученный познанием», или, более подробно, «система суждений с принципиальной и единой организацией, обоснованная на объективной закономерности».
- 4) Знания—это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области [5].
- 5) Знания—это хорошо структурированные данные, или данные о данных (метаданные) [5].
- 6) Знания формализованная информация, на которую ссылаются или используют в процессе логического вывода [18].
- 7) Знания это обобщенная и формализованная информация о свойствах и законах предметной области, с помощью которой реализуются процессы решения задач, преобразование данных и самих знаний и которая используется в процессе логического вывода [2, 4, 10].
- 8) Знания это основные закономерности предметной области, позволяющие человеку решать конкретные производственные, научные и другие задачи, т. е. факты, понятия, взаимосвязь, оценки, правила, эвристики (иначе, фактические знания), а также стратегии принятия решения в этой области (иначе, стратегические знания).
- 9) Под знанием системы искусственного интеллекта мы понимаем совокупность информации о внешнем мире, о мире абстрактных наук и о знаниях разумных существ, т. е. об их внутреннем мире, включая знания о внутреннем мире самой системы. Каждая из указанных компонент знания выражается в виде описания объектов, отношений между ними и законов, действующих в соответствующем мире [10].
- 10) Знание это «проверенный практикой результат познания действительности, верное ее отражение в мышлении человека».

Отражение существующего мира в мышлении человека связано с процессом абстрагирования, который заключается в выделении наиболее существенных свойств и признаков явлений или объектов, наблюдаемых в окружающей действительности, и представлении их в такой упрощенной форме, которая необходима для логических умозаключений и принятия эффективных решений. Таким образом, мы приходим к модели, так как указанное упрощенное представление окружающей действительности и есть модель предметной области. Человек, познавая окружающий мир, строит модели процессов, явлений, предметов — объектов реального мира. Модель является информационным эквивалентом части реального мира (предметной области) и лежит в основе процесса познания. В инженерной практике

именно модели ассоциируются со знанием, но понятие «знание» шире понятия «модель», хотя знания, в основном, состоят из моделей. Знания включают в себя не только сведения об объектах реального мира, но и информацию о механизмах выведения новых знаний на основе имеющихся.

- 1) Представление это действие, делающее некоторое понятие воспринимаемым посредством фигуры, записи, языка или формализма. Теория знаний изучает связи между субъектом (изучающим) и объектом. Знание (в объективном смысле) то, что известно (то, что знаем после обучения).
- 2) Представление знаний это соглашение о том, как описывать реальный мир. В естественных и технических науках принят следующий традиционный способ представления знаний. На естественном языке вводятся основные понятия и отношения между ними. Но при этом используются ранее определенные понятия и отношения, смысл которых уже известен. Далее устанавливается соответствие между характеристиками (чаще всего, количественными) понятий знаний и подходящей математической моделью. Основная цель представления знаний строить математические модели реального мира и его частей, для которых соответствие между системой понятия проблемного знания может быть установлено на основе совпадения имен переменных модели и имен понятий без предварительных пояснений и установления дополнительных неформальных соответствий.
- 3) Представление знаний формализация истинных убеждений посредством фигур, записей или языков. В ИИ особый интерес уделяется формализациям, воспринимаемым (распознаваемым) ЭВМ. Возникает вопрос о представлении знаний в памяти ЭВМ, т.е. о создании языков и представления знаний. Они преобразуют наглядное представление (созданное посредством речи, изображением, естественным языком, вроде английского или немецкого, формальным языком, вроде алгебры или логики, рассуждениями и т.д.) в пригодное для ввода и обработки в ЭВМ. Результат формализации должен быть множеством инструкций, составляющих часть языка программирования. На современном этапе развития экспертных систем используется несколько форм представления знаний.

Понятно, что выбор методов описания предметной области в значительной степени определяет возможности системы. В настоящее время предложено много подходов к представлению знаний в ИИ, однако эта проблема все еще далека от разрешения.

По-видимому, универсальным языком представления знаний является естественный язык. Однако его использование в этом качестве в информационных системах невозможно, т.к. не существует алгоритмов обработки естественно-языковых текстов.

Основными проблемами при создании СИИ являются неэффективные средства представления знаний, недостаточная вложенность семантики и формализмов представления. Для того, чтобы точнее это уяснить, необходимо обратиться к формам представления знаний (моделям).

4) Представление знаний обычно выполняется в рамках той или иной системы представления знаний. Системой представления знаний (СПЗ)

называют средства, позволяющие описывать знания о предметной области с помощью языка представления знаний, организовывать хранение знаний в системе (накопление, анализ, обобщение и организация структурированности знаний), выводить новые знания из имеющихся, находить требуемые знания, устранять устаревшие знания, осуществлять интерфейс между пользователем и знаниями.

8.2 Данные и знания в интеллектуальных системах

Представление данных и знаний это одна из компонент интеллектуальной системы, основным понятием которой они и являются.

Данные — это факты, характеризующие объекты, а также их свойства.

Под данными мы понимаем двоичные слова фиксированной длины, информационные структуры, абстрактные типы данных, структура которых задается программистом. Исторически, понятие «данные» видоизменялось и усложнялось, и появилась концепция знаний. Представим свойства знаний, присущие этой форме представления знаний [2–4, 9–13].

1) Внутренняя интерпретируемость знаний.

Переход от физических адресов к относительным, а позднее к символьным, автоматическое распределение физической памяти позволило формировать ЭВМ таблицу соответствий, в которой относительным и символьным адресам соответствовали физические адреса, назначаемые ЭВМ. Но имена информационных единиц не подвергались анализу, и машина не могла ответить на запрос о содержимом ее памяти, если программист не составил для ответа программу. При переходе к знаниям в память машины вводится информация о некоторой протоструктуре информационных единиц, т. е. специальное машинное слово, в котором указано, в каких разрядах хранятся данные. Атрибуты могут играть роль имен для машинных слов, которые соответствуют строкам таблицы. Каждая строка является экземпляром протоструктуры. Свойство внутренней интерпретируемости всех информационных единиц обеспечивается СУМД (система управления манипулирования данными), хранимой в реляционной БД.

2) Структурированность.

Данные должны обладать гибкой структурой. Внутренняя структура сначала как совокупность разрядов была со временем усложнена рекурсивной вложенностью одних информационных единиц в другие. Каждая информационная единица может быть включена в состав другой, и из каждой такой единицы можно выделить составляющие ее элементы.

Таким образом, появилась возможность произвольного установления отношений типа «элемент-множество», «тип-подтип», «род-вид», «часть-целое», «элемент-класс», отражающих характер их взаимосвязи.

Самостоятельные части информационной единицы называют обычно слотами, а полная запись ее имеет вид (имя информационной единицы <имя 1-го слота, значение 1-го слот>...<имя п-го слота, значение п-го слота>). Иногда в представлении знаний вместо термина «слот i-го уровня» используют специальные названия. Например, части слота 1-го уровня называются ячейками, а их части — фасетами.

3) Связность.

Между информационными единицами устанавливаются связи (внешняя структура) различного типа. Отношения могут быть каузальными (причинно-следственными), «аргумент-функция» (вычисление функций), структуризации (иерархия информационных единиц), пространственные и временные, а также отношения, определяющие порядок выбора информационных единиц из памяти машины, совместимость и несовместимость отдельных событий или фактов в некоторой ситуации, расположенных в одной области пространства, и т. д. Связи позволяют строить процедуры анализа знаний на совместимость, противоречивость и другие, которые трудно реализовать при хранении традиционных массивов данных.

Например, в общей модели представления знаний — семантической сети (иерархической сети, в вершинах которой находятся информационные единицы, а дуги соответствуют различным связям) иерархические связи определяются отношениями структуризации, а неиерархические связи — отношениями иных типов. Иерархические связи представляются отношениями ISA (is an instance of), связывающими элементы (примеры) класса с самим классом и АКО (а kind of), использующими для связи одного класса с другим.

Так, индивидуальный объект «Иванов А. Д.» связывается ISA-отношением с классом «группа 446», который, в свою очередь, связывается АКО-отношением с более общим классом «факультет ФСУ».

4) Семантическая метрика.

На множестве информационных единиц полезно задавать отношения, характеризующие ситуационную близость этих единиц. Это отношение релевантности информационных единиц, которые позволяют находить знания, близкие к имеющимся, т. е. сузить пространство поиска нужной информации и обнаруживать знания, близкие к уже найденным. Для примера можно выделять в информационной базе некоторые типовые ситуации или учитывать частоту появления тех или иных ситуаций или конкретных представлений в типовых ситуациях.

В декларативной форме представления знаний семантические и синтаксические знания в определенной мере отделены друг от друга, что придает большую универсальность и общность.

В процедурной форме представления знаний семантика объекта включается в описание элементов базы знаний, что обеспечивает более быстрый поиск решений по сравнению с декларативными, однако уступает им в возможности накопления и актуализации знаний (рис. 8.1).

5) Шкалирование.

Отдельные информационные единицы могут соотноситься между собой с помощью различных шкал.

Метрические шкалы (абсолютные и относительные) устанавливают количественные соотношения и порядок тех или иных совокупностей информационных единии

Порядковые шкалы устанавливают лишь порядок информационных единиц. На них нет никакой метрики, и мы не можем количественно оценить, насколько одно значение превосходит другое.

Среди порядковых шкал выделяют нечеткие порядковые шкалы (лингвистические

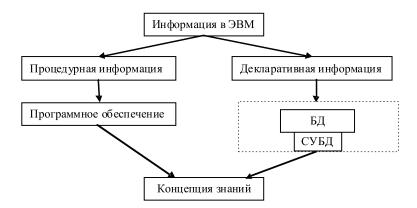


Рис. 8.1 – Классификация информации в ЭВМ.

шкалы). Шкала ограничена с двух сторон маркерами «никогда или 0» и «всегда или 1», т. е. частота появления события равна 0 в первом случае и 1 во втором. Остальным маркерам типа «очень редко», «редко», «не часто» и т. д. соответствуют не точечные деления, а интервалы, величина и расположение которых зависит от интерпретации словесных оценок, относящихся к шкале.

Оппозиционные шкалы. Образуются с помощью пар слов-антонимов: «хороший-плохой», «белый-черный», «здоровый-больной» и т. д. Это бинарная оппозиция типа противопоставления друг другу двух фактов, явлений или свойств.

6) Активность.

Практика программирования показала, что программы и данные, с которыми эти программы работают, отделены друг от друга. В программе сосредоточено процедурное знание. Оно хранит в себе информацию о том, как надо действовать, чтобы получить нужный результат. Машина выступает как калькулятор, а программа как активатор данных.

Декларативные знания хранят в себе информацию о том, над чем надо выполнять эти действия. Процедурное знание формирует обращение к декларативному знанию, воплощенному на первых этапах развития программирования в пассивно лежащие в памяти машины данные.

Для интеллектуальных систем складывается противоположная ситуация. Не процедурные знания активируют декларативные, а наоборот, та или иная структура декларативных знаний оказывается активатором для процедурных знаний. Появление в базе фактов или описаний событий, установление связей может стать источником активности системы. Совокупность средств, обеспечивающих работу со знаниями образует систему управления базой знаний (СУБЗ).

Сравнение технологий разработки приложений представлено на рис. 8.2

Вместе с тем, приведенная характеристика знаний в ИС не является совершенно изолированной от того, что мы обычно понимаем под знанием. Такие черты, как внутренняя интерпретируемость, структурированность, связность, семантическая метрика, и активность, присущи любым более или менее крупным блокам человеческих знаний, и в этом смысле знания в компьютерной системе можно рассматривать как модель или образ того или иного фрагмента человеческого знания.

Однако связь знаний в специфическом для ИИ смысле со знанием в более привычном, «обычном» смысле не ограничивается лишь сходством некоторых струк-

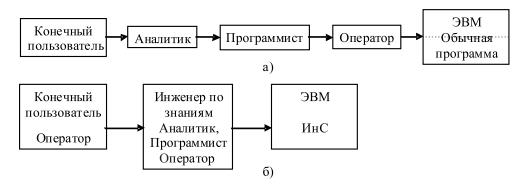


Рис. 8.2 — Сравнение двух технологий разработки приложения: a — классическая технология разработки программного обеспечения; δ — технология разработки программного обеспечения с использованием ИнС.

турных характеристик. Ведь значительная часть информации, представляемой в базе знаний ИС, есть ни что иное, как знания, накопленные в той области, где должна применяться данная система. Исследование этого знания под углом зрения задач построения ИС — определяет технологический подход ИИ к знанию как таковому, под которым понимается следующее: формализация знаний и модель знаний как алгоритм решения задачи.

Центральным вопросом построения систем, основанных на данных и знаниях, является представление, т. е. формализация и моделирование. Обобщенную схему представления данных и знаний представим на рис. 8.3.



Рис. 8.3 – Обобщенная схема представления данных и знаний.

Модели представления данных. Различают три основные модели представления данных: реляционные, сетевые и иерархические. В настоящее время появилось второе поколение таких моделей, в рамках которых происходит постепенное слияние данных и знаний. В развитых моделях представления данных можно выделить два компонента: интенсиональные представления и экстенциональные представления. Оба компонента хранятся в базе данных. При этом в ее экстенциональную часть входят конкретные факты, касающиеся предметной области, а в интесиональные представления описывают конкретные объекты, а интенсиональные представления фиксируют те закономерности и связи, которым все эти конкретные объекты, события, явления и процессы обязаны в данной проблемной области удовлетворять. Экстенсиональные представления относятся к данным, относительно интенсиональных — единого мнения нет. Апологеты баз данных говорят в этом случае

о схемах базы данных, а представители искуственного интеллекта — о знаниях проблемной области.

В сетевых моделях представления данных используются табличные и графовые представления. Вершинам графа соответствуют информационные единицы, называемые записями, которые представляют собой специальным образом организованные таблицы, а дугам графа — типы отношений между записями. В отличие от реляционных моделей, в которых запрещается повторение одинаковых строк во множестве, определяющем экстенциональное отношение, в сетевых моделях такие повторения допустимы. Другим отличием является явное задание отношений между записями с помощью графа, а не опосредованное, как в реляционной модели.

В иерархических моделях представления данных, в отличие от сетевых и реляционных моделей, схема базы данных представляет собой множество схем отношений, упорядоченных в древообразную структуру. В отличие от сетевой модели, в которой экстенциональные отношения могут быть агрегированными частями (т. е. представлять собой целые фрагменты графа), в иерархических моделях (как и в реляционных) элементами доменов могут быть лишь простые факты.

В базах данных выделяют две составляющие: язык описания данных (ЯОД) и язык манипулирования данными (ЯМД). Средства ЯОД ориентированы, с одной стороны, на то, как на физическом уровне ЭВМ будут представляться данные, интенсиональные и экстенсиональные отношения, а с другой стороны, на семантику проблемной области, так как в них входят операции по классификации, обобщению и агрегированию экстенсиональных и интенсиональных отношений. Эти средства позволяют вводить обобщенные атрибуты и записи, устанавливать новые схемы отношений на интенсиональном уровне и строить многоуровневые иерархии на множествах обобщенных атрибутов. Таким образом, ЯОД позволяет реализовать в базах данных такие особенности знаний, которые ранее назывались структурированностью и связностью.

На сегодняшний день можно считать завершенной теорию реляционных баз данных, сейчас уже ясно, что все три типа баз данных равномощны.

Второе поколение баз данных характеризуется рядом особенностей моделей представления данных. Расширение возможностей таких моделей происходит изза ослабления требований к виду отношений в экстенциональных представлениях и к виду отношений в интенсиональных представлениях.

Семантические сети являются наиболее общей моделью представления знаний, так как в них имеются средства для выполнения всех пяти требований, предъявляемых к знаниям. Но такая универсальность семантических сетей имеет и негативную сторону. Если допускать в них произвольные типы отношений и связей, не являющиеся отношениями в математическом смысле (например, ассоциативные связи), то сложность работы с таким образом организованной информацией резко возрастает. Поэтому в базах данных второго поколения вводятся ограничения на характер структур и типов информационных единиц, находящихся в семантической сети, и на характер связей, задаваемых дугами.

Модели представления знаний. В ИС используются различные способы описания знаний [16].

1) Логические модели. В основе моделей такого типа лежит формальная система, задаваемая четверкой вида $M = \langle T, P, A, B \rangle$. Множество T есть мно-

жество базовых элементов различной природы, например слов из некоторого ограниченного словаря, деталей детского конструктора, входящих в состав некоторого набора, и т. п. Важно, что для множества T существует некоторый способ определения принадлежности к этому множеству. Процедура такой проверки может быть любой, но за конечное количество шагов она должна давать положительный или отрицательный ответы на вопрос, является ли х элементом множества T. Для знаний, входящих в базу знаний, можно считать, что множество образуют все информационные единицы, которые введены в базу знаний извне, а с помощью правил вывода из них выводятся новые производные знания. Это свойство логических моделей делает их притягательными для использования баз знаний. Оно позволяет хранить в базе лишь те знания, которые образуют множество, а все остальные знания получать из них по правилам вывода.

- 2) Сетевые модели. В основе моделей этого типа лежит конструкция, названная ранее семантической сетью. Сетевые модели можно формально задать в виде $H < I, C_1, C_2, \dots, C_n, \Gamma$. Здесь I — множество информационных единиц, C_1, C_2, \ldots, C_n — множество типов связей между информационными единицами; отображение Г задает между информационными единицами, входящими в І, связи из заданного набора типов связей. В зависимости от типов связей, используемых в модели, различают классифицирующие сети, функциональные сети и сценарии. В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами. Функциональные сети характеризуются наличием функциональных отношений. Их часто называют вычислительными моделями, так как они позволяют описывать процедуры «вычислений» одних информационных единиц через другие. В сценариях используются казуальные вычисления, а также отношения типов «средство-результат», «орудие-действие» и т. п. Если в сетевой модели допускаются связи различного типа, то ее обычно называют семантической сетью.
- 3) Продукционные модели. В моделях этого типа используются некоторые элементы из логических и сетевых моделей. Из логических моделей за-имствована идея правил вывода, а из сетевых моделей описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам сетевого описания происходит трансформация семантической сети за счет смены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. Таким образом, в продукционных моделях процедурная информация явно выделена и описывается иными средствами, чем декларативная информация. Вместо логического вывода, характерного для логических моделей, проявляется вывод на знаниях.
- 4) Фреймовые модели. В отличие от моделей других типов во фреймовых моделях фиксируется жесткая структура информационных единиц, которая называется протофреймом. Значением слота может быть практически что угодно (числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов). В качестве знания слота может выступать

набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать «принцип матрешки». При конкретизации фрейма ему и слотам присваиваются конкретные имена и происходит заполнение слотов. Таким образом, из протофреймов получаются фреймы-экземпляры. Переход от исходного протофрейма к фрейму-экземпляру может быть многошаговым за счет постепенного уточнения значений слотов. Связи между фреймами задаются значениями специального слота с именем «Связь». Часть специалистов по ИС считает, что нет необходимости специально выделять фреймовые модели в представлении знаний, так как в них объединены все основные особенности моделей остальных типов.

8.3 Понятийная структура предметной области

При построении описания выделенного фрагмента реальной действительности (предметной области — Π O) одним из основных допущений является предположение о том, что мир состоит из независимых объектов (предметов, процессов, вещей или сущностей), которые могут быть уникально идентифицированы. В дальнейшем при обозначении реальных объектов Π O для однозначности будем использовать термин «сущность».

Напомним, что термин «проблемная область» включает понятие «предметная область» (т. е. множество сущностей, описывающих область экспертизы) плюс решаемые в ней задачи (представляемые в виде исполняемых утверждений типа правил, процедур, формул и т. д.).



Сущность— это объект произвольной природы, принадлежащий реальному или воображаемому (виртуальному) миру.

............

.....

На самом общем уровне сущности делятся на следующие категории (классы): предмет, свойство (атрибут), состояние, процесс, событие, оценка, модификатор, квантификатор, модальность.

Сущность ПО необходимо адекватным образом представить в памяти вычислительной машины на основе базовых понятий и правил получения сложных синтаксических конструкций.

Понятие (концепт) — класс сущностей, объединяемых на основе общности призначных (атрибутивных) структур.

Простое понятие — это тройка, состоящая из имени, интенсионала и экстенсионала понятия.

Имя понятия — любой идентификатор; интенсионал понятия — множество атрибутов (свойств) понятия с областями их определения; экстенсионал понятия — совокупность кортежей значений, удовлетворяющих интенсионалу; атрибуты — первичные характеристики данной ПО, не подлежащие дальнейшей структуризации.

Сложные понятия— это понятия, образованные из ранее определенных понятий применением некоторых правил.

Напомним, что классические модели данных (иерархические и сетевые) базируются на таких понятиях, как запись, атрибут и связь. Современные модели данных (реляционные и семантические) используют математическое понятие отношения, которое задается на множествах, и понятие объекта для представление сущностей ПО в БД.

Структура признаков — это совокупность взаимосвязанных существенных признаков (атрибутов), характеризующих понятие, выделяющее его среди других понятий.

Совокупность существенных признаков, характеризующих понятие, называется его содержанием (интенсионалом).

Класс сущностей, объединяемых в понятие, называется его объемом (экстенсионалом). Сущности, входящие в объем понятия и называемые экземплярами, вариантами или фактами, задаются перечислением признаков.

Знаковое представление понятий. Понятия именуются с помощью слов или словосочетаний естественного языка, которые играют роль знаков или имен.

Сущности ПО, на которые можно ссылаться с помощью имени или знака, называются денотатами.

Денотат знака — это конкретная сущность или предмет, на которые указывает данный знак. Другими словами, денотат знака является способом интерпретации данного знака в рамках некоторой ситуации, рассматриваемой в ПО. Денотат знака - это значение, которое может иметь знак в рамках данной знаковой ситуации [15]. Можно указать на следующие основные свойства знаковой ситуации:

- имена способны замещать денотаты. Например, имя понятия АВТОМО-БИЛЬ может использоваться в качестве заместителя любого конкретного автомобиля;
- имя нетождественно денотату, оно не может полностью заменить денотат. Так, при алгебраических преобразованиях мы можем использовать буквенные обозначения чисел. Однако если необходимо вычислить выражение, то требуется подставить конкретные числа (денотаты) вместо букв;
- связь «имя-денотат» многозначна, т. е. некоторое имя может обозначать множество денотатов омонимия, и, наоборот, одному денотату можно поставить в соответствие несколько имен синонимия.

Отношение, связывающее обозначаемую сущность (denotat), отражающее ее понятие (consent) и его имя (designat), графически выражается треугольником Фреге (рис. 8.4). Сплошными линиями обозначен базовый треугольник Фреге, характеризующий однозначное отношение. Он является желательным, но не всегда достижим. Пунктирными линиями изображены два треугольника, представляющие два случая неоднозначности отношения.

Треугольник «сущность 1, понятие 1, имя 2» в сочетании с базовым иллюстрирует случай синонимии. Сущности 1 и понятию 1 ставятся в соответствие два имени, называемые синонимами.

Треугольник «сущность 2, понятие 2, имя 1» в сочетании с базовым иллюстрирует случаи омонимии и полисемии. Сущностям 1 и 2 и понятиям 1 и 2 ставится в соответствие одно и то же имя, что свидетельствует о его многозначности. В случае полисемии понятия 1 и 2 характеризуются общим происхождением. При

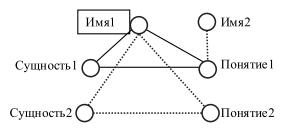


Рис. 8.4 – Треугольники Фреге, отражающие неоднозначность решения.

омонимии понятия 1 и 2 имеют различное происхождение.

Схемы и формулы понятий. Как уже отмечалось, сущности, составляющие объем понятия, различаются с помощью признаков. Признаковые отношения предписывают одним понятиям выполнять роль некоторых качественных свойств по отношению к другим понятиям. Признаки понятий могут быть отнесены к одному из следующих типов: дифференциальные, характеристические и валентные [5].

Дифференциальные признаки используются в качестве характеристики содержания понятия.

Характеристические — это признаки, которые позволяют отличить сущности, относящиеся к объему одного и того же понятия.

Валентные — это признаки, обеспечивающие связь между различными понятиями. Без потери общности можно считать такие связи бинарными.

Признак характеризуется именем и значением. Можно выделить несколько типов значений признаков: логические, числовые, символьные и др. Имя признака вместе с его значением образует полное наименование соответствующего признака. Например, признак ВОЗРАСТ вместе со значением «30 лет» образует наименование признака «возраст 30 лет».

Совокупность имен дифференциальных, характеристических и валентных признаков составляет схему понятия, обозначаемую как $shm\ P$. Таким образом, схему понятия можно представить в виде тройки

$$shm P = \langle D, H, V \rangle, \tag{8.1}$$

где $D = \{D_i\}, i = 1, n$ — множество имен дифференциальных признаков;

 $H = \{H_i\}, j = 1, m$ — множество имен характеристических признаков;

 $V = \{V_k\}, k = 1, l$ — множество имен валентных признаков.

Имена признаков будем обозначать прописными буквами латинского алфавита (возможно, с индексами) A, B, C, D, \ldots , а значения — строчными.

Тот факт, что признак A_i данного события принимает одно из возможных значений $a_j^i \in dom\,A_i$, будем выражать в виде пары $A_ia_j^i$. Здесь $dom\,A_i$ обозначает множество (домен) всех возможных значений признака A_i . Тогда каждая сущность e, принадлежащая объему понятия P, может быть представлена в виде множества пар дифференциальных, характеристических и валентных признаков.

Любое понятие можно выразить в виде логической формулы, в которой элементы, образующие понятия, представлены предикатами и соединены логическими операциями конъюнкции, дизъюнкции и отрицания. В зависимости от вида этой формулы представляемые ею понятия можно классифицировать как простые и сложные:

1) простое понятие P(X) представляется логическим выражением

$$P(X) < -\&_{i=1}^n A_i\left(z_i^i, X\right),\,$$

где $z_j^i \in dom A_i - j$ -е значение признака A_i , а $A_i \left(z_j^i, X \right)$ предикат истинный, если $z_j^i E \, dom A_i$ признак A_i сущности X принимает значение z_j^i , иначе — ложный.

Для данного понятия в логической формуле может присутствовать не более одного значения некоторого признака A_i . Это логическое выражение допускает простую интерпретацию: любая сущность X, обладающая совокупностью признаков $z_j^i \in dom A_i$, является денотатом данного понятия P(X). Примером простого понятия может служить понятие КНИГИ-МАРТИНА:

КНИГИ-МАРТИНА (Название, Y)<-АВТОР (Название, Мартин) & ГОД-ИЗ-ДАНИЯ (Название, 1982) & ЯЗЫК (Название, Русский), & ЦЕНА (Название, Y), которое описывает множество книг, написанных Дж. Мартином и выпущенных на русском языке в 1982 г. Денотатами понятия КНИГИ-МАРТИНА являются книги «Организация баз данных в вычислительных системах» и «Сети ЭВМ». В качестве дифференциальных признаков выступают признаки (АВТОР, Мартин) и (ГОД-ИЗДАНИЯ, 1982), а в качестве характеристического признака—(ЦЕНА, Y).

Экстенсионал и интенсионал понятия. В семиотике для характеристики понятия и знаковой ситуации получили распространение категории денотата и концепта, а в логике для тех же целей служат термины «экстенсионал» и «интенсионал».

Одно и то же понятие может обозначать разные денотаты. Множество всех объектов, предметов или сущностей, являющихся денотатами понятия, составляет объем, или экстенсионал, понятия. Если экстенсионал понятия P обозначить через extP, то может записать следующее выражение [46]:

$$ExtP = [e_1, e_2, \dots, e_n],$$

где $[e_1, e_2, \dots, e_n]$ — сущности, являющиеся денотатами понятия P.



Экстенсионал понятия — это совокупность всех его допустимых денотатов, соответствующих концепту этого понятия. Так, чтобы описать экстенсионал понятия АВТОМОБИЛЬ, следует рассмотреть класс всех автомобилей. Мы полагаем, что в БЗ используются только конечные экстенсионалы понятия.

и ограничиться понятиями, которые допускают не более одного значения

Если ограничиться понятиями, которые допускают не более одного значения некоторого признака, то между экстенсионалом понятия и его признаками может быть установлена функциональная связь вида

$$f_i: extP \to dom A_i$$

сопоставляющая каждой сущности $e^i_j \in ext\,P$ некоторое значение признака $z^i_j \in ext\,P$

Разные понятия могут иметь один и тот же экстенсионал. Так, понятия «множество точек, равноудаленных от сторон данного угла» и «прямая, делящая угол

на два равных угла», имеют одно и то же множество точек в качестве денотатов (биссектрису данного угла). Эти понятия имеют общий экстенсионал, выражают различный смысл, который мы связываем с их интенсионалом. Следовательно, понятие не полностью характеризуется экстенсионалом. Необходимо учитывать также интенсиональный аспект понятия, который связан с его концептом.



Интенсионал понятия— это тот смысл, который мы вкладываем в данное понятие, т. е. интенсионал характеризует концепт данного понятия, его содержание. Интенсионал понятия Р будем обозначать int P.

.....

Связи между различными категориями, используемыми для описания понятий в логике и семиотике, представлены на рис. 8.5



Рис. 8.5 – Связи между категориями, используемыми для описания понятий.

Суммируя вышеизложенное, можно представить понятие в виде тройки:

$$P = \langle int P, ext P, shm P \rangle$$
.

В зависимости от характера деятельности внимание проектировщиков ИнС концентрируется на различных составляющих понятия: системные аналитики и администраторы задач приложений, создающие формальную модель ПО, больше должны опираться на интенсионал понятий ПО, а администраторы базы данных — на их схему. Тем самым понятие становится чрезвычайно удобным средством, которое позволяет, с одной стороны, путем использования интенсионала выразить семантические отношения для некоторого фрагмента реального мира, а с другой стороны, с помощью схемы обеспечить возможность перехода к менее детальному описанию и представлению этой информации в базе знаний.

Абстрагирование понятий. Абстракция понятий представляет собой основной механизм, с помощью которого человек познает окружающий мир.



Абстракция— это выделение существенных признаков и связей понятий, используемых при решении данной задачи и игнорировании несущественных.

Абстрагирование является основным методологическим приемом при разработке программного обеспечения. Оно помогает разбить решаемую задачу на подзадачи, каждая из которых проще исходной задачи. Причем при рассмотрении начальной задачи нет необходимости учитывать те детали и ту более подробную информацию, которые используются на этапе рассмотрения подзадач. В поле зрения должны находиться только те сведения, которые позволяют охватить проблему целиком и осуществить ее декомпозицию на более простые подзадачи.

Абстрагирование обеспечивает упорядочение, структуризацию и понимание информации о реальном мире. Поэтому методы абстракции широко используются в формальном описании предметной области.



Известны следующие основные типы абстрагирования понятий: агрегация, обобщение, типизация и ассоциация [15].

......

.....

Перечисленным типам абстрагирования соответствуют определенные логические приемы, с помощью которых может быть достигнут необходимый результат. К таким приемам образования понятий обычно относят:

- синтез и анализ понятий;
- объединение понятий на основе их сходства и подобия;
- сравнение и сопоставление конкретных сущностей с целью выявления общих признаков;
- связывание двух или более понятий.

8.4 Модели знаний

8.4.1 Логические модели

Формальные системы. В основе логических моделей представления знаний лежит понятие формальной системы [1, 2, 10, 19]. Формальная система задается четверкой:

$$M = \{T, P, A, R\},\$$

где T — множество базовых элементов;

P — множество синтаксических правил;

A — множество аксиом;

R — множество правил вывода.

Любая логическая система (теория) и исчисление предикатов, в частности, может быть построена на базе как синтаксических, так и семантических концепций.

Теорию, построенную на базе семантических концепций, называют теорией моделей, а теорию, построенную на базе синтаксических концепций, — аксиоматической теорией. При обоих способах построения некоторой теории необходимо определить понятия алфавита и формулы.

Элементы множества T — алфавит формальной системы. Это некоторое счетное множество символов теории.

Произвольные конечные последовательности символов алфавита называются выражениями теории.



Формулой теории называется некоторое выделенное подмножество выражений теории.

Алфавит исчисления предикатов состоит из следующего множества символов.

Знаков пунктуации: (),.

Пропозиционных связок: ~, ∨, ∧, →.

Знаков кванторов: ∀, ∃.

Символов переменных: $x_k, k = 1, 2, ...$

n-местных(размерных) функциональных букв $f_k^n, k \ge 1, n \ge 0, f_k^0$ называют константными буквами.

n-местных предикатных букв (символов) $P_k^n, k \geqslant 1, n \geqslant 1.$

Обычно вместо x_k пишут u, v, w, x, y, z вместо $f_k^0 - a, b, c, d$; вместо $f_k^n, n \neq 0 - f, g, h$, а вместо $P_k^n - P, Q, R, S, T, V, W$.

Из символов алфавита можно строить различные выражения. Выделим среди них те, которые представляют для нас интерес.

1) Термы.

Каждый символ переменной или константной буквы является термом.

Если $t_1, \ldots, t_n, n \ge 1$ г— термы, то и $f_k^n(t_1, \ldots, t_n)$ гявляется термом.

Выражения являются термом только в том случае, если это следует из правил а) или б).

2) Элементарные формулы (атомы).

Если P_k^n г— предикатная буква, а t_1, \ldots, t_n г— термы, то $P_k^n(t_1, \ldots, t_n)$ г— элементарная формула (атом).

3) Формулы или правильно построенные формулы (ППФ).

Всякая элементарная формула есть формула.

Если D и B — формулы, и x — переменная, то каждое из выражений (~ D), ($D \lor \lor B$), ($D \land B$), ($D \to B$), ($\forall \lor D$), ($\exists \lor D$) гесть формула.

Выражение является формулой только в том случае, если это следует из правил а) и б).

Итак, множество синтаксических правил P позволяет строить из элементов T синтаксически правильные совокупности базовых элементов. На множество синтаксических правил не накладываются особые ограничения. Требуется только, чтобы существовала конструктивная процедура, которая позволяла бы за конечное число шагов дать однозначный ответ на вопрос, является ли данная совокупность элементов из T синтаксически правильной. Такие совокупности называются $\Pi \Pi \Phi$.

Среди всех ПП Φ выделяют некоторое подмножество аксиом A. При этом должна существовать процедура, позволяющая для любой ПП Φ решить вопрос, является ли она аксиомой данной формальной системы.

Наконец, множество R— это конечное множество отношения между ППФ, называемых правилами вывода (см. часть 1 главу 3). Правило вывода — это отношение на множестве формул. Например, если из формул F_1, F_2, \ldots, F_n непосредствен-

но выводится формула F, то это можно записать в виде

$$\frac{F_1,F_2,\ldots,F_n}{F},$$

где F_1, F_2, \ldots, F_n называются посылками правила, а F — его следствием (заключением).

Применяя правила вывода к множеству аксиом A, можно получить новые ППФ, к которым можно опять применить правила из R. Это позволяет осуществить вывод новых ППФ.

Множество R называют также множеством семантических правил. А множество $\Pi\Pi\Phi$, полученное после применения правил к аксиомам, называется множеством семантически правильных совокупностей.

Выводом формулы B из формул A_1, A_2, \ldots, A_n называется последовательность ППФ F_1, F_2, \ldots, F_m , такая, что $F_m = B$, а для любого $i(l \le i \le m)$ формула F_i есть либо аксиома формальной системы, либо одна из исходных формул A_1, A_2, \ldots, A_n , либо непосредственное следствие формул $F_1, F_2, \ldots, F_{i-1}$ полученное с помощью правил вывода. Некоторая ППФ F является выводимой в формальной системе (является теоремой), если существует вывод, в котором последней формулой является F. Сокращенно вывод F из F_1, F_2, \ldots, F_n , будем записывать в виде $F_1, F_2, \ldots, F_n \vdash F$. Если имеется эффективная процедура, позволяющая по данной ППФ F устанавливать, существует ли ее вывод в формальной системе, то данная формальная система называется разрешимой, в противном случае — неразрешимой.

Интересной стороной формальной системы с точки зрения представления знаний является то, что ее можно рассматривать как генератор новых знаний. В этом случае из множества аксиом A, представляющих собой знания, изначально хранящиеся в базе знаний СИИ, выводятся с помощью правил вывода производные знания [21]. Для этого машина вывода анализирует внутреннее представление знаний в памяти ЭВМ. Выделяемые в памяти ЭВМ цепочки символов, представляющие конкретные знания о предметной области, называют образами, а цепочки символов, соответствующие посылкам правил, — образцами. Если образ и образец сопоставимы, то в базу знаний добавляется новый элемент знаний — заключение правила.

Рассмотрим два класса формальных систем: исчисление высказываний и исчисление предикатов.

Исчисление высказываний. Высказыванием называется предложение, содержание которого можно оценить как истинное или ложное. Среди формул высказываний выделяют формулы, которые истинны во всех интерпретациях. Данные формулы называются тавтологиями или общезначимыми формулами. Формула исчисления высказываний называется противоречием, если она принимает значение ЛОЖЬ во всех интерпретациях. Две формулы называются равносильными (эквивалентными), если они принимают одинаковые значения на всех наборах входящих в них переменных (интерпретациях).

Процесс дедуктивного вывода базируется на понятии логического следования. Формула B является логическим следствием формулы A, если B истинна на всех наборах значений переменных (интерпретациях), на которых истинна A. Можно показать, что $A \Rightarrow$ если и только если импликация $A \rightarrow B$ является тавтологией, т. е.

 $\models A \to B$. (\models означает Истина). Понятие логического следствия можно обобщить на совокупность высказываний:

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \Rightarrow B$$

Можно также показать, что формула B является логическим следствием последовательности формул A_1, A_2, \ldots, A_n тогда, и только тогда, когда формула $A_1 \wedge A_2 \wedge \ldots \wedge A_n \sim B$ является противоречием; что и представлено в части 1 главе 3.

Множество правил вывода R в исчислении высказываний задается двумя правилами: правилом подстановки и правилом заключения и одним дополнительным — правилом силлогизма. Применение правил вывода для получения следствий опирается на монотонность исчисления высказываний.

В заключение отметим, что исчисление высказываний - это разрешимая система. Это следует из теоремы о полноте исчисления высказываний, которая утверждает, что в исчислении высказываний выводима любая общезначимая ППФ. Значит, если рассматриваемая формула общезначима, то она выводима в исчислении высказываний. Если это не так, то формула не принадлежит множеству семантически правильных формул.

Исчисление предикатов. Исчисление высказываний характеризуется ограниченными возможностями для представления знаний. Объясняется это тем, что в исчислении высказываний рассматриваются логические связи только между утверждениями, а сама структура утверждений не анализируется. Поэтому в исчислении высказываний нельзя установить связь между тем, о чем идет речь (объект), и тем, что сообщается о данном объекте (предикат). Исчисление предикатов рассматривает логические связи между различными элементами утверждений.

Предикатом называется неоднородная двузначная логическая функция от любого числа аргументов. Ее записывают в виде $P(x_1, x_2, ..., x_n)$ и называют п-местным предикатом. Здесь аргументы $x_1, x_2, ..., x_n$ принадлежат одному и тому же или нескольким различным множествам, представляющим объекты предметной области. Сами аргументы называют предметными переменными, а их конкретные значения — предметными постоянными.

В алфавит исчисления вводятся знаки кванторов: \forall , \exists . Для кванторов справедливы законы де Моргана:

$$\sim (\forall x P(x)) = \exists x \sim P(x);$$

$$\sim (\exists x P(x)) = \exists x \sim P(x).$$

Если область определения предиката P(x)гконечна и $X = \{a, b, c\}$ гто высказывание $\forall x P(x)$ гэквивалентно конъюнкции $P(a) \land P(b) \land P(c)$ га высказывание $\exists x P(x) -$ дизъюнкции $P(a) \lor P(b) \land P(c)$.

Формула называется выполнимой (непротиворечивой), если она истинна, по крайней мере в одной интерпретации. Формула называется опровержимой, если она ложна, хотя бы в одной интерпретации.

Отношения равносильности и логического следования определяются как в главе 6.

Для доказательства общезначимости формул исчисления предикатов используется аппарат логики исчисления высказываний, дополненный аксиомами и прави-

лами вывода выражений, содержащих кванторы. Рассматривая исчисление предикатов как формальную систему, отметим, что к системе аксиом исчисления высказываний добавляются еще две аксиомы:

$$\forall x P(x) \rightarrow P(t)$$
,

где t — терм;

$$P(t) \rightarrow \exists x P(x).$$

В этих аксиомах P(x) — любая формула, содержащая свободное вхождение аргумента x, причем ни одно из них не находится в области действия квантора по t. Формула P(t) получается из P(x) заменой всех свободных вхождений x на t.

Правила вывода исчисления предикатов. К правилам пропозиционной логики добавляются следующие правила, построенные на основе рассмотренных аксиом.

Правило универсальной конкретизации (УК):

$$\frac{\forall x P(x)}{P(t)}.$$

Правило экзистенциальной конкретизации (ЭК):

$$\frac{\exists x P(x)}{P(a)}$$
.

Правило экзистенциального обобщения:

$$\frac{P(a)}{\exists x P(x)}.$$

Правило универсального обобщения:

$$\frac{a \to P(x)}{a \to \forall x P(x)}.$$

Процесс поиска состоит в сопоставлении посылок правил вывода с утверждениями, содержащимися в базе знаний. В исчислении предикатов доказательство выводимости формулы из совокупности посылок сводится, как и в исчислении высказываний, к установлению общезначимости или противоречивости формулы (см. 7.1).

Исчисление предикатов — неразрешимая формальная система, т. е. не существует эффективной процедуры, позволяющей узнать по данной формуле, возможен ли ее вывод в исчислении предикатов или нет (теорема А. Черчя, 1936 г.). Формула выводима, если она общезначима. Следовательно, исчисление предикатов первого порядка неразрешимо относительно общезначимости. Иными словами, не существует общего метода для установления общезначимости любых формул исчисления предикатов первого порядка. Тем не менее если на самом деле некоторая формула общезначима, то для нее существует процедура проверки общезначимости. Поэтому исчисление предикатов — полуразрешимая формальная система.

8.4.2 Семантические сети

Историческая справка. Фрег представил логические формулы в виде деревьев, которые однако мало напоминают современные семантические сети. Еще одним пионером стал Чарльз Сандерз Пирс, который использовал графические записи в органической химии.

Он сформулировал правила вывода с использованием экзистенциоанальных графов. В психологии Зельц использовал графы для представления наследственности некоторых характеристик в иерархии концептов. Научные изыскания Зельца имели огромное влияние на изучение тактики в шахматах. Он, в свою очередь, повлиял на таких теоретиков, как Саймон и Ньюэлл.

Что касается лингвистики, то первым ученым, занимавшимся разработкой графических описаний, стал Теньер. Он использовал графическую запись для своей грамматики зависимостей. Теньер оказал огромное влияние на развитие лингвистики в Европе.

Впервые семантические сети были использованы в системах машинного перевода в конце 50-х-начале 60-х годов. Первая такая система, которую создала Мастерман, включала в себя 100 примитивных концептов, таких, как, например, НАРОД, ВЕЩЬ, ДЕЛАТЬ, БЫТЬ. С помощью этих концептов она описала словарь объемом 15000 единиц, в котором также имелся механизм переноса характеристик с гипертипа на подтип.

В системах искусственного интеллекта семантические сети используются для ответа на различные вопросы, изучения процессов обучения, запоминания и рассуждения. В конце 70-х сети получили широкое распространение. В 80-х годах границы между сетями, фреймовыми структурами и линейными формами записи постепенно стирались. Выразительная сила больше не является решающим аргументом в пользу выбора сетей или линейных форм записи, поскольку идеи, записанные с помощью одной формы записи, могут быть легко переведены в другую. И наоборот, особо важное значение получили второстепенные факторы, как читаемость, эффективность, так же учитываются легкость введения в компьютер, редактирование и распечатка.

Общие положения и виды семантических сетей. Термин «семантическая» означает «смысловая», а сама семантика — это наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. наука, определяющая смысл знаков [3].

А. В. Андрейчиков в своей книге пишет, что общепринятого определения семантической сети не существует. Хотя большинство авторов определяют ее как систему знаний некоторой предметной области, имеющую определенный смысл в виде целостного образа сети, узлы которой соответствуют понятиям и объектам, а дуги — отношениям между объектами [2]. Или как представление предметной области в виде ориентированного графа, вершинами которого выступают понятия, а ребрами — связи между ними [3, 5].

Семантическую сеть понятий образует совокупность взаимосвязанных понятий. Эта сеть является концептуальной моделью ПО (предметной области). Обычно она состоит из понятий различных категорий: объектов, свойств, операций, событий и т. д.

Семантическая сеть понятий содержит в первую очередь интенсиональные знания (знания об абстрактных объектах) о ПО. Если ПО рассматривать как совокупность понятий и связей (отношений) между ними, то семантические сети дают возможность представлять знания о ПО в наглядной и структурированной форме, что не всегда возможно при других способах представления знаний. Если в сети предусмотрена иерархия обобщения, то при этом можно использовать механизм логического вывода, основанный на наследовании признаков. Например, начиная навигацию от исходного понятия, можно по ребрам связей достичь другого понятия за некоторое число шагов. Однако такая навигация может стать простым перебором вариантов, если путь продвижения по сети понятий не определен.

Семантическую сеть можно рассматривать как композицию троек (синтагм или триплетов) вида ArB, где A и B—два понятия, а r—связь между ними.

Связь между понятиями семантической сети выражает минимальный объем знаний, простейший факт, относящийся к двум понятиям. Более сложные утверждения в рамках семантической сети могут быть определены путем выделения соответствующих подграфов. В естественном языке такие подграфы выделяются с помощью отдельных законченных предложений, которые описывают определенные ситуации, возникающие между понятиями ПО [5].

В качестве понятий обычно выступают абстрактные или конкретные объекты, а отношения—это связи типа: «это» («AKO—A-Kind-Of», «is»), «имеет частью» («has part»), «принадлежит», «любит». Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс элемент класса (цветок роза);
- свойство значение (цвет желтый);
- пример элемента класса (роза чайная).

Можно предложить несколько классификаций семантических сетей, связанных с типами отношений между понятиями.

По количеству типов отношений:

- однородные с единственным типом отношений.
- неоднородные с различными типами отношений.

По типам отношений:

- бинарные отношения связывают два объекта.
- n-арные отдельные специальные отношения связывают более двух понятий.

Наиболее часто в семантических сетях используются следующие отношения:

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

- логические связи (И, ИЛИ, НЕ);
- лингвистические связи и др.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети, отражающей поставленный запрос к базе.

На рис. 8.6 изображена семантическая сеть. В качестве вершин тут выступают понятия «Человек», «Иванов», «Волга», «Автомобиль», «Вид транспорта», «Двигатель».

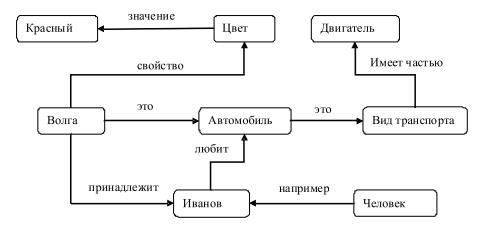


Рис. 8.6 – Семантическая сеть.

Данная модель представления знаний была предложена американским психологом Куиллианом. Основным ее преимуществом является то, что она более других соответствует современным представлениям об организации долговременной памяти человека (Скрэгг).

Недостатком этой модели является сложность организации процедуры поиска вывода на семантической сети [3].

В книге А.В. Андрейчикова семантические сети Куиллиана систематизируют функции отношений между понятиями с помощью следующих признаков:

- множество-подмножество (типы отношений «абстрактное конкретное», «целое часть», «род вид»);
- индексы (свойства, имена прилагательные в языке и т. п.);
- конъюктивные связи (логическое И);
- дизъюнктивные связи (логическое ИЛИ);
- связи по Исключающему ИЛИ;
- отношения «близости»;
- отношения «сходства различия»;
- отношения «причина следствие» и др.

При построении семантической сети отсутствуют ограничения на число элементов и связей. Поэтому систематизация отношений между объектами в сети необходима для дальнейшей формализации. Пример семантической сети представлен на рис. 8.7 [2].

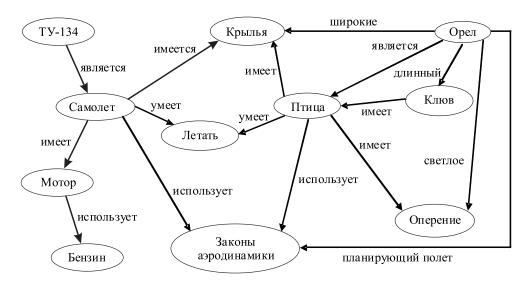


Рис. 8.7 – Семантическая сеть, показывающая взаимоотношения птицы и самолета.

Д. В. Гаскаров внимание уделяет предметной области, ведь ПО в каждый момент времени может быть представлена в виде совокупностей сущностей, понятий и ситуаций. Выделенная совокупность сущностей, понятий и ситуаций ПО называется ее состоянием. Любое изменение состояния ПО связано с некоторым событием в ПО. Основной сферой использования событий являются ситуации, приводящие к изменению состояния понятий ПО.

Каждой ситуации можно поставить в соответствие некоторое утверждение или суждение об ее истинности или ложности, поэтому взаимосвязь группы понятий носит информационный характер, что позволяет считать ситуации основной категорией для описания ПО. Различие методов концептуального моделирования в основном определяется теми формальными средствами, которые используются для описания ситуаций: в семантических сетях и фреймах — это понятия и их взаимосвязи.

Использование ситуаций обеспечивает механизм для декомпозиции семантической сети понятий, моделирующей ПО, на более простые, семантически связанные блоки понятий.



Ситуация — это простейшая логически законченная структура, которая может выступать в виде некоторого автономного блока семантической сети понятий, рассматриваемой ПО.

Для реализации семантических сетей существуют специальные сетевые языки: NET, язык реализации систем SIMER+MIR и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний: PROSPECTOR, CASNET, TORUS[2].

Важные черты системы SIMER+MIR представлены в книге Гаспарова Д. В. В этой системе вершины (узлы) сети делятся на *события*, *атрибуты*, *комплексы признаков* и *процедуры*.

Под событиями здесь понимаются различные объекты ПО: суждения, факты (индивидные понятия), результаты наблюдений, рекомендации. События могут представляться как словосочетаниями, так и числами. События группируются тематически или функционально в *разделы*. Имена последних должны отличаться от имен входящих в них событий. Одно событие может быть более чем в одном разделе.

События в системе делятся на *характеризуемые* и *характеризующие* (события-признаки). Например, событие «Дождливая погода» характеризуется событием «Идет дождь». Это событие называется *обусловленным* признаком первого, ибо дождливая погода без дождя *невозможна*.

В зависимости от направления влияния на событие признаки делятся на *по- пожительные* и *отрицательные*. В примере с дождем приведен положительный признак. Примером отрицательного признака является «Сухая земля».

Характеризующее событие, имеющее несколько значений, называется *атрибутом*. Например, свойством понятия «Время года» является «Погода». Поскольку последняя имеет несколько значений: «Холодная», «Теплая», «Дождливая», — ее считают атрибутом понятия «Время года».

Несколько признаков могут объединяться в комплекс, характеризующий событие в большей степени, чем отдельный признак.

Если существование события возможно только при реализации всех его свойств, оно относится к *полным* событиям. Полным, в частности, является событие, имеющее единственный атрибут или комплекс признаков.

Процедуры являются специфическими компонентами сети, выполняющими преобразование информации. Они позволяют вычислять значения одних атрибутов на основании других, оперируя как с числами, так и с символами.

Для вывода знания события в сетевой модели делятся на *исходные* (признаки) и *целевые* (гипотезы). Значения признаков предполагаются известными. Все признаки, помимо присущих им значений: Истинно (Да) и Ложно (Нет), — имеют еще два стандартных значения: Пока неизвестно и Неизвестно. При задании последнего значения признак исключается из рассмотрения. Значения исходных атрибутов либо выбираются из определенного списка, либо вводятся извне.

Объектами вывода в рассматриваемой модели являются гипотезы. К ним относятся рекомендации, диагнозы, прогнозы и другие решения, определяемые спецификой ПО. Условием вывода должно быть существование хотя бы одной гипотезы. В этом случае решением является оценка ее истинности [5].

Виды семантических связей. Семантическая связь (СС) отражает отношение понятий в понятийной системе. В лексике им соответствуют лексемы любого вида, в том числе представляющие *предикаторы* «меньше», «равно», «если, то» и др.

Внелексические свойства СС выражаются через рефлексивность, симметричность и транзитивность. Обозначим их значения следующим образом: Rf — рефлексивность; Nrf — нерефлексивность; Arf — антирефлексивность (ни одной рефлексии); Sm — симметричность; Ns — несимметричность; Ans — антисимметричность (ни одной симметрии); As — асимметричность (контекстное свойство — обращение связи дает иную связь из списка); Tr — транзитивность; Ntr — нетранзитивность.

Относительно сочетания перечисленных свойств СС делятся на типы, представленные в табл. 8.1.

класса	Тип связи (Х, Ү)	Каноническая форма	Свойство
1	C	V	AuC No No
1	Gen — генератив-	Х является элементом Ү	Arf, Ns, Ntr
	ная	Х находится в ситуации Ү	Arf, As, Tr
	Sit — ситуативная	Х отрицает Ү	Arf, Sm, Ntr
2	Neg — негативная	Y.	NI C NI NI
2	Ins — инструмен- тальная	Х является средством Ү	Nrf, Ns, Ntr
3	Com — комитатив-	Х сопровождается Ү	Rf, Ans, Tr
	ная	Х иногда увеличивает воз-	Rf, Sm, Ntr
	Cor — коррелятив-	можность Ү	
4	Fin — финитивная	Х является целью Ү	Arf, Ns Ntr
	Cous — каузальная	Х вызывает Ү	Nrf, Ns, Tr
	Рот — потенсивная	Х может вызывать Ү	Nrf, Ns, Ntr
	иочено $\rightarrow Nrf$.		
	лючено $\rightarrow Nrf$. $\rightarrow Arf$.	Пример	
Невозмо Вегетат: Этвет 1 для	ожно $\rightarrow Arf$. ивные расстройства с Com.	сопровождаются вегетативны	
Невозмо Вегетата Этвет 1 для Симмет	ожно $\rightarrow Arf$. швные расстройства с Com. причность определяет	сопровождаются вегетативны	и: объекты <i>А</i> и <i>В</i> :
Невозмо Вегетат Ответ 1 для Симмен яются мес ожения. П	ожно $\rightarrow Arf$. швные расстройства с Сот. причность определяет стами $(Arb \rightarrow BrA)$, и	сопровождаются вегетативныеся по критерию перестановки выясняется справедливостивете высказыванию приписы	и: объекты <i>А</i> и <i>В</i> в ъ полученного пр
Невозмо Вегетаті Ответ 1 для Симмен іяются мес	ожно $\rightarrow Arf$.	сопровождаются вегетативныеся по критерию перестановки выясняется справедливостивете высказыванию приписы	и: объекты А и В г ъ полученного пр
Невозмо Вегетатт Ответ 1 для Симмен инстивном противном	ожно $\rightarrow Arf$.	сопровождаются вегетативныеся по критерию перестановки выясняется справедливостивете высказыванию приписы	и: объекты A и B в полученного пр вается свойство S

Таблица 8.1 – Свойства семантических связей.



Com. Это соответствует свойству Ns.

Свойство Ns уточняется на более *сильные* свойства: Ans и As. Первое имеет место для любых примеров анализируемой связи. Например, для связи Com имеет место свойство Ans.



Для выявления свойства As используется критерий *обращения*: если высказывания *А предикатор В* и *В предикатор А* принадлежат различным типам высказываний в таблице, то имеет место свойство As.

......

Транзитивность выявляется на основе критерия *трансформации*, в высказывание вводится уточнение (как 2-я посылка).

Например, в качестве уточнения в высказывание вводится модальность необходимости: $ecnu\ A$, то необходимо появится B. При справедливости такого высказывания оно относится к каузальному типу (Caus), в противном случае (модальность возможности) — к потенсивному типу (Pot).

Типы связей в высказываниях разбиты в табл. 8.1 на четыре группы. Группы 1–3 отражают *одномоментные* зависимости, группа 4—разномоментную зависимость между объектами высказывания (следствие реализуется позже посылки).

Типы связей внутри групп 1–3 различаются свойством рефлексии. Внутри этих групп типы связей различаются по свойствам симметрии.

Поскольку в группе 4 все связи обладают одинаковым свойством несимметрии, для их различения используются свойства рефлексивности и транзитивности.



Таким образом, свойства *одномоментности*, *рефлексивности*, *симметричности* и *транзитивности* можно использовать в качестве признаков установления типов высказываний.

Перево вырода типов связей с перенислении ими свойствами в канестве основа-

Дерево вывода типов связей с перечисленными свойствами в качестве оснований деления изображено на рис. 8.8.

И возвращаясь к системе SIMER+MIR, в ней устанавливаются двусторонние связи между двумя событиями, причем они могут различаться. Реализованы следующие связи.

При наблюдении события A может наблюдаться (обычно наблюдается) событие B. Эта связь является положительной. Она порождает событие B в качестве гипотезы или увеличивает уверенность в его истинности.

При наблюдении события A всегда наблюдается событие B. Эта положительная связь более сильная.

При наблюдении события A обычно отсутствует (может отсутствовать) событие B. Эта связь является отрицательной. Она уменьшает уверенность в истинности события B.

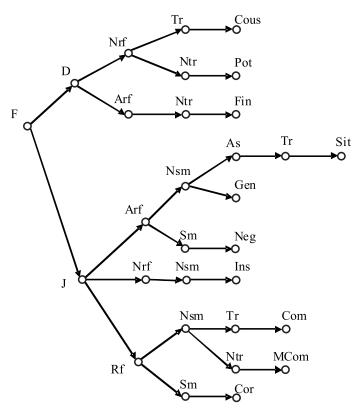


Рис. 8.8 – Дерево выводов типов высказываний.

При наблюдении события A всегда отсутствует событие B. Эта связь называется исключающей, поскольку событие B исключается из списка гипотез [5].

Методы обобщения знаний на сетях. Применение аппарата семантических сетей для решения задач классификации и формирования понятий позволяет получать обобщенные представления множеств объектов, имеющих наглядную семантическую трактовку, и использовать в процессе обобщения семантику таких понятий, как «признак», «имя», «класс», «отношение». Роль языка представления наблюдений и классов (понятий) играет в данном случае выбираемый формализм семантической сети.

Рассмотрим методы *обобщения на сетях*. Пусть Z- множество объектов l. Каждый объект представляется сетью, называемой семантическим графом, который включает вершины двух типов: объектные и предикатные. Объектной вершине приписывается имя (объекта), имя базового класса объекта и вектор его признаков; предикатной вершине — имя отношения, выполняющегося между объектами (возможно с отрицанием). Семантический граф, предназначенный для представления объекта, $l \in Z$, распадается на иерархически упорядоченное множество двухуровневых р-графов, служащих для представления объектов l, его «частей» (р — потомков), частей его частей и т. д.; р-подграф содержит на первом уровне объектную вершину Vs, связанную отношением р (целое — часть) с объектными вершинами второго уровня, которые связаны между собой отношениями через предикатные вершины.

Таким образом, модели и методы обобщения знаний можно представить в виде схем (рис. 8.9) [5].



Рис. 8.9 – Классификация методов и моделей обобщения знаний.

Объекты и отношения в семантических сетях. Семантические сети не являются однородным классом моделей представления знаний. Часто общей основой отнесения схемы представления знаний к семантической сети является то, что она представляется в виде направленного графа, вершины которого соответствуют объектам (понятиям, сущностям) предметной области, а дуги — отношениям (связям) между ними. И узлы, и дуги, как правило, имеют метки (имена). Имена вершин и дуг обычно совпадают с именами соответствующих объектов и отношений предметной области.

Объекты предметной области, отображаемые в семантической сети, можно условно разделить на три группы: обобщенные, индивидные (конкретные) и агрегатные объекты.

Обобщенный объект соответствует некоторой собирательной абстракции реально существующего объекта, процесса или явления предметной области. Например, «изделие», «предприятие», «сотрудник» и т. д. Обобщенные объекты фактически представляют определенные классы предметной области.

Индивидный объект — это каким-то образом выделенный единичный представитель (экземпляр) класса. Например, «сотрудник И. Н. Петров».



Агрегатным называется составной объект, образованный из других объектов, которые рассматриваются как его составные части.

Например, изделие состоит из совокупности деталей, предприятие состоит из совокупности отделов, служб, цехов.

Введенная классификация объектов является относительной. В зависимости от решаемой задачи один и тот же объект может рассматриваться как обобщенный или индивидный, как агрегатный или неагрегатный.

Типы связей между объектами семантических сетей могут быть любыми. Но чаще всего применяются следующие основные связи (отношения): «род-вид», «является представителем», «является частью». Наличие связи типа «род-вид» между обобщенными объектами A и B означает, что понятие A более общее, чем понятие B. Любой объект, отображаемый понятием B, отображается и понятием A, но не наоборот. Например, понятие «животное» — это родовое понятие для объекта «птица». Все свойства родового объекта A, как правило, присущи и видовому объекту B. Иными словами, объект B наследует свойства объекта A.

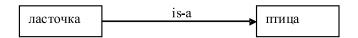
Связь «является представителем» существует обычно между обобщенным и индивидным объектом, когда индивидный объект выступает в роли представителя некоторого класса. Так, индивидный объект «овчарка Альма» является представителем (экземпляром) обобщенного объекта «овчарка». Экземпляр может быть представителем нескольких обобщенных объектов, что соответствует множественному наследованию.

В ряде случаев между связями «род-вид» и «является представителем» не делают различий, отмечая, что эти связи задают отношение «общее-частное» Иногда это приводит к недоразумениям. Поэтому для формализации таких связей будем использовать отношения ako (от англ. a-kind of — разновидность) и is_a (от англ. is a member of the class — быть представителем класса).

Не менее важно отношение «является частью» (англ. part of). Данное отношение связывает агрегатный объект с его составными частями. Оно позволяет отражать в базе знаний структуру объектов предметной области. Иногда данное отношение обозначают меткой «имеет» (англ. has) [6].

В книге Х. Уэна, Т. Коямы, Т. Окамота, Б. Мацуби, М. Исудзуки [18] рассматривается более подробно отношение is_a и part_of. Эти типы предикатов отношений существуют в иерархической структуре понятий. Иерархия показывает отношение включения понятия. Например, в предложении «человек is_a млекопитающее» (А human is a mammal) главной мыслью является, что человек принадлежит к классу млекопитающих. Это означает, что в данном предложении используется отношение включения или отношение совпадения. Способ включения можно назвать понятием верхнего уровня, а способ удаления — понятием нижнего уровня. Экземпляр (instance) нижнего уровня содержит в основном все атрибуты, которые имеет экземпляр понятия верхнего уровня (прототип). Это свойство называется наследованием атрибутов между уровнями иерархии is a.

Выражение part_of показывает отношение «целое-часть». Например, в предложении «Нос есть часть тела» экземпляр «нос» является частью любого экземпляра «тело». Этот способ показывает отношения между экземплярами класса, причем основная часть показывает внутреннюю структуру предиката. Рассмотрим теперь предложение, представляющее факт «все ласточки — птицы». Это предложение можно представить в следующей форме, использовав для этого две вершины, обозначающие «ласточка» и «птица», и дугу, показывающую отношение между ними:

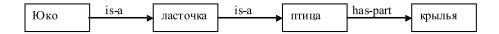


Если далее присвоить некоторое имя ласточке, например «Юко», то сеть можно расширить следующим образом:



В данном случае вместе с тем, что с помощью такой сети представлены два факта «Юко — ласточка» и «ласточка — птица», из нее, используя отношение is_a, можно вывести факт «Юко — птица». Этот факт показывает, что способ представления семантической сетью позволяет легко делать выводы благодаря иерархии наследования.

Семантическими сетями можно также представлять знания, касающиеся атрибутов объекта. Например, факт «птицы имеют крылья» отображается в следующем виде:



Это означает, что, используя отношения is_a и part_of, можно вывести факт «Юко имеет крылья». Другими словами, факт, объявляемый для вершин на верхнем уровне иерархической структуры, на основе предпосылки, говорящей о справедливости его для узлов нижнего уровня, показывает возможность вывода множества фактов с помощью отношения is_a. Вывод такого типа называется наследованием свойства, а ветвь is_a называется ветвью наследования свойства.

Вершины семантической сети обычно показывают объект проблемной области, концепт, ситуацию и т. п., а дуги — отношения между ними. При расширении семантической сети в ней возникают другие отношения. Например, если приведенную выше сеть дополнить фактом «Юко владеет гнездом», то можно построить следующую сеть:

где гнездо-і— это конкретное гнездо, которым владеет Юко; оно является экземпляром понятия «гнездо». Другими словами, выше приведено представление с обобщенными отношениями совокупности предметов между ветвью отношения владения, гнездом и гнездом-і. Кроме того, при желании можно дополнить сеть информацией «Юко владеет гнездом с весны по осень», тогда вершинами необходимо представить не только объекты, но и ситуации и действия. На рис. 8.10 показана полученная таким образом семантическая сеть, в частности показано, что для вершины ситуации (владеть-і) определено несколько связей. Такая вершина называется падежной рамкой (case frame), она определяет различные аргументы предиката ситуации.

Большинство систем с семантическими сетями имеет унифицированную структуру применительно к факторам действия и объекта по отношению к некоторому концепту. Преимущества использования такой структуры в вершинах сети заключаются в возможности наследования ожидаемых значений и значений по умолчанию, которые являются значениями атрибута в вершине экземпляра типа «владееті», аналогичного показанному на рис. 8.10.

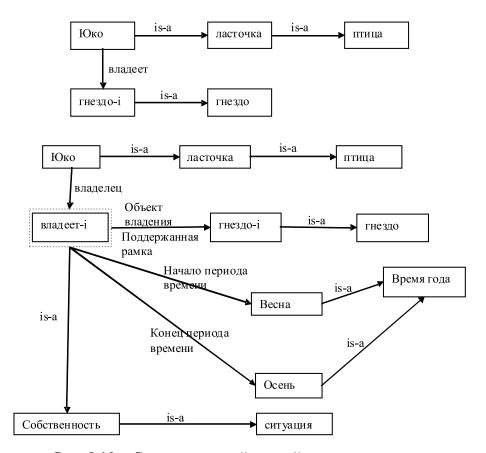


Рис. 8.10 – Сеть с падежной рамкой и наследованием.

Итак, большой проблемой, характерной для семантических сетей, является наследование атрибутов между иерархическими уровнями. Другими словами, результат вывода, получаемого с помощью семантической сети, не гарантирует достоверность как логический формализм. Это обусловлено тем, что процедура вывода по определению не более как наследование ветви is_a. Например, в сети, показанной на рис. 8.11, при дополнении факта «ласточка—это вид, который подвергается опасности» фактом «виды, подвергающиеся опасности, изучаются натуралистами» можно исходя из последнего факта вывести заключение, что ласточка Юко изучается натуралистами. Однако это заключение не обязательно является правильным. Вследствие этого требуются такие способы представления данных и вывода, которые обеспечивали бы одновременно управление наследованием [12].

В книге В. Н. Бондарева многообразие объектов сети подразделяется на три группы:

- объекты-понятия сведения о физических и абстрактных объектах предметной области;
- объекты-события абстрактные или конкретные действия, которые могут привести к изменению состояния предметной области;
- объекты-свойства уточняют понятия и события, например указывают характеристики понятий (цвет, форму, размеры и т. п.), фиксируют параметры событий (место, время, продолжительность).

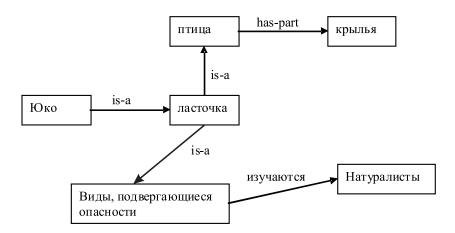


Рис. 8.11 – Проблема наследования по ветвям is-a.

А многообразие отношений, используемых в семантических сетях, подразделяется на следующие группы:

- лингвистические отношения, которые подразделяются на падежные (агент, объект, инструмент, время, место), глагольные (наклонение, время, вид, число, залог) и атрибутивные (цвет, размер, форма);
- логические отношения (конъюнкция, дизъюнкция, отрицание, импликация);
- теоретико-множественные отношения (множество подмножество, видподвид, целое-часть, элемент множества);
- квантифицированные отношения, которые подразделяются на логические кванторы общности и существования, нечеткие кванторы (много, несколько, часто).

Таким образом, рассмотренные ранее отношения далеко не исчерпывают всего набора отношений, применяемых в семантических сетях. Но они образуют хорошую основу для построения прикладных баз знаний.

Формализация семантической сети. При построении семантической сети отсутствуют ограничения на число связей элементов и свойств и сложность сети. Поэтому функции, указываемые указателями, желательно в некоторой степени упорядочить. Куиллиан и др. предложили определять функции между понятиями с помощью (1) подмножества — старшего множества, (2) индекса (наречие, имя прилагательное), (3) логического правила И, (4) логического правила ИЛИ, (5) логического правила Исключающее-И, (6) логического правила Исключающее-ИЛИ и т. п. Кроме того, в качестве операторов отношения для группировки элементов (вершин) были предложены отношения «близости», «следствия», «предпосылки» и «сходства». Поскольку все эти отношения зависят от мира, в котором они используются, то и их унификация является сложной задачей. Внимания заслуживают отношения, предложенные для многих миров. Только благодаря этим отношениям становится возможным представление семантическими сетями.

Важность модели семантической сети Куиллиана с точки зрения многочисленных приложений определяется следующими моментами.

В отличие от традиционных методов семантической обработки с анализом структуры приложения были предложены новые парадигмы в качестве модели

представления структуры долговременной памяти, в которой придается значение объему языковой активности.

Был предложен способ описания структуры отношений между фактами и понятиями с помощью средства, называемого семантической сетью, отличающейся несложными представлениями понятий.

Была создана реальная система, осуществлено моделирование человеческой памяти и разработана технологическая сторона концепции понимания смысла [8].

Способы описания семантических сетей и логический вывод. После знакомства с общими понятиями семантических сетей рассмотрим способы описания семантических сетей. Наиболее часто для этих целей используют концептуальные графы Дж. Соува и блочные структуры Г. Хендрикса. Блочные структуры описаны в книге Уэна X.

Г. Хендрикс предложил метод, называемый разделением семантической сети, и ввел понятие иерархически упорядоченного множества пространств, определяющих границы действия вершин экземпляров (рассматриваемых в качестве переменных). Например, в следующем факте «собака укусила почтальона» слова «собака», «укус» и «почтальон» обозначают соответствующие классы. Если ввести вершины переменных, отображающие специальные понятия в виде экземпляров «с», «к», «п», то можно построить сеть, как показано на рис. 8.12, а.

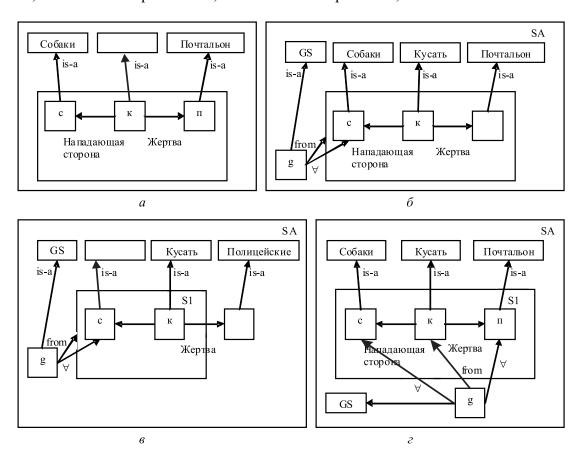


Рис. 8.12 – Разделение семантической сети.

С помощью такой сети можно представить следующий факт: Каждая собака кусает почтальона

Если этот факт проиллюстрировать с помощью логической формулы, то получим:

$$\forall x$$
 собака (x) ($\exists y$ почтальон $(y) \cap$ кусать (x,y)).

Результат разделения этого факта можно представить в виде сети, изображенной на рис. 8.12, 6. В данном случае вершина g представляет собой приведенное выше утверждение, а GS — это $\forall x$ собака (x). Поэтому g является экземпляром GS. Экземпляр GS (в данном случае g) имеет два атрибута — Form и несколько символов \forall . Form указывает утверждение, которое нужно сделать, а \forall соответствует квантору общности. В данном примере внимания заслуживает то обстоятельство, что другие переменные «к» и «п» разделены отношением Form, ограничены лишь квантором существования.

Посмотрим, как правильно выполнить квантование переменных с помощью разделения сети. Например, предложение:

Все городские собаки кусают полицейских;

можно представить в виде семантической подсети, показанной на рис. 8.12, в. Поскольку в данном случае вершина «п», которая означает жертву, существует вне подсети (не является точно заданной переменной), то она не зависит от значения «с» и показывает некоторого полицейского.

Далее рассмотрим следующий пример:

Каждая собака кусает каждого почтальона

Семантическая подсеть в данном случае выглядит как показано на рис. 8.12, г. Вершина п ограничивает с помощью дуги Form элемент S1, кроме того, из него выходят еще две дуги null, соединяющие его с любой вершиной «собака» и любой вершиной «почтальон».

Важное значение в семантических подсетях имеет отношение включения пространства. В примере, показанном на на рис. 8.12, *г*, пространство S1 содержится в пространстве SA, и для поиска S1 из SA допускается использовать только дугу Form. Кроме того, из вершины «с» можно выйти на вершину «собака», но обратный ход через ту же дугу is_а неосуществим. Это связано с тем, что пространство S1 находится на более низком уровне, чем SA.

Хендрикс использовал метод разделения семантической сети в системе понимания речи, где для условного обозначения правил, определяющих категорию объекта, применялись пространства. В классификационных системах, основанных на этих категориях, на уровне категории посредством запоминания совместно используемых данных осуществляется сжатие избыточной информации. Правила категорий в сетях, определяющих значения атрибутов, связанных с различными классами глаголов, поддерживают транслирующие программы. Это связано с необходимостью определения, как правильно выполнять семантический разбор, а также с необходимостью объединения внешней структуры с соответствующим сетевым представлением. В семантических подсетях представлены такие средства, как семантическая структура, структура разговора, структура ответов [22].

Теперь рассмотрим описание семантических сетей в виде концептуальных графов. Вершинами концептуального графа являются либо объекты (понятия, сущности) предметной области, либо концептуальные отношения. Ребра концептуального графа связывают между собой вершины-понятия и вершины-отношения. При этом ребра могут исходить из вершины-понятия и заканчиваться в вершинеотношении, и наоборот. На рис. 8.13 вершины «мяч» и «красный» соответствуют

вершинам-понятиям, а вершина «цвет» — бинарному отношению. Для того, чтобы различать указанные типы вершин, на концептуальных графах их изображают прямоугольником и эллипсом соответственно [10].

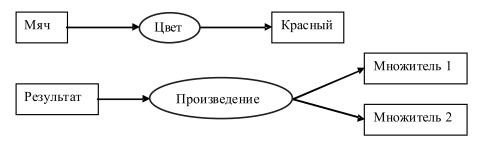


Рис. 8.13 – Примеры концептуальных графов.

Одно из преимуществ выделения отношений в самостоятельные вершины концептуального графа заключается в упрощении представления n-арных отношений. Такое отношение представляется вершиной-отношением с n ребрами.

Обычно каждый концептуальный граф фиксирует одно предложение. Тогда база знаний будет представляться в виде совокупности графов. Граф на рис. 8.14 отражает более сложное предложение «Иван закрепил деталь стула клеем».

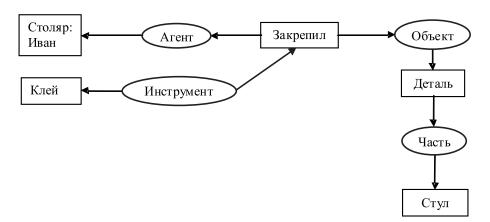


Рис. 8.14 – Концептуальный граф предложения.

Этот граф использует падежные отношения глагола «закрепить» и демонстрирует возможности концептуальных графов для представления предложений естественного языка.

Каждая вершина-понятие концептуального графа может иметь метку типа. Тип обозначает класс принадлежности вершины. На концептуальном графе метку типа вершины отделяют двоеточием от конкретного имени вершины. На концептуальных графах можно также вводить индивидные вершины-понятия с одинаковыми именами. Чтобы различать экземпляры понятий, используется числовой маркер, перед которым стоит символ #. Кроме числового маркера, на концептуальных графах могут применяться обобщенные маркеры, обозначаемые *. В сочетании с переменной, записываемой после этого знака, обобщенный маркер оказывается полезным, если две различные вершины представляют один и тот же объект (рис. 8.15).

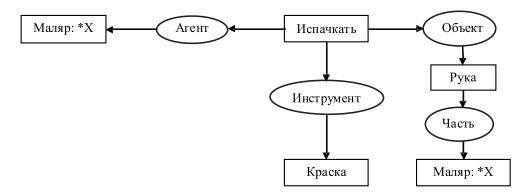


Рис. 8.15 – Концептуальный граф с обобщенным маркером.

На концептуальных графах вводятся операции, которые позволяют выполнять их преобразование. Новые графы получаются с помощью четырех операций.

Копирование. Позволяет сформировать из исходного графа его копию.

Специализация. Заключается в замене обобщенной вершины ее индивидным (конкретным) вариантом. При этом, если вершина отмечена обобщенным маркером, то его заменяют конкретным индивидным маркером; тип вершины можно заменить соответствующими подтипами, обладающими необходимыми свойствами.

Объединение. Позволяет получать из двух графов, содержащих хотя бы одну идентичную вершину, один, переориентируя в одном из графов связи на другой граф.

Упрощение. Исключает на графах дубликаты отношений. При этом исключается вершина-отношение и связанные с ней дуги. Дублирование отношений часто возникает при выполнении объединения.

Операция специализации позволяет сопоставлять две вершины концептуального графа и при успешном сопоставлении выполнять объединение. Совместное использование операций специализации и объединения обеспечивает реализацию механизма наследования. Например, заменив обобщенный маркер индивидным, мы распространяем свойства типа на конкретный индивидный объект. Если в ходе выполнения операции ограничения происходит замена типа на его подтип, то имеет место наследование «класс — подкласс». Операции специализации и объединения конкретизируют исходный граф. Если граф G_1 — конкретизация графа G_2 , то верно и то, что G_2 — обобщение G_1 . Обобщение широко используется при обучении СИИ, позволяя выводить правдоподобные умозаключения по частным примерам.

Рассмотренные выше простейшие операции не гарантируют получения правильных утверждений на основе преобразованных графов. Конкретизация вершинпонятий, выполняемая в ходе операций специализации и объединения, не всегда соответствует действительности. Например, если в ходе выполнения операции объединения соединяются два графа, агенты которых обозначены одним и тем же понятием, то это вовсе не означает, что может случиться так, что эти понятия соответствуют двум различным индивидным объектам. Например, предложения «Собака грызет кость» и «Собака находится в парке» можно объединить, если речь идет об одном животном, а не о двух разных. Тем не менее рассмотренные операции не приводят к утверждениям, лишенным смысла, и позволяют выполнять

правдоподобные рассуждения.

Концептуальные графы могут содержать вершины, которые являются предложениями. Такие вершины изображаются в виде прямоугольного блока, содержащего подграф, соответствующий предложению. В качестве примера на рис. 8.16 изображен граф утверждения «Алексей полагает, что Татьяне нравится игрушка».

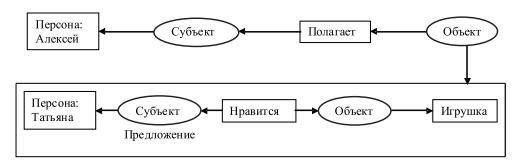


Рис. 8.16 – Концептуальный граф, содержащий предложение.

Введение отдельных блоков для обозначения предложений позволяет фиксировать отношения между предложениями. В приведенном примере глагол «полагает» представляет не конкретное физическое действие, а выражает ощущения Алексея. Поэтому на графе введено новое отношение «субъект», представляющее лицо, испытывающее те или иные ощущения. Утверждения, содержащие подобные лексические единицы (доверять, верить, намериваться и т. п.), являются объектом исследований модальной логики.

Концептуальный граф позволяет естественным образом выражать конъюнкцию. Например, с помощью концептуального графа мы легко можем представить предложение «Собака находилась в саду и грызла кость». Сложнее обстоит дело с другими логическими операциями: дизьюнкцией, логическим отрицанием, кванторами.

Для представления отрицания применяется унарный логический оператор «Не». Оператор применяется к предложению. Утверждение, фиксируемое предложением, считается в этом случае ложным. Граф на рис. 8.17 соответствует предложению «Нет игрушек, которые нравятся Татьяне».

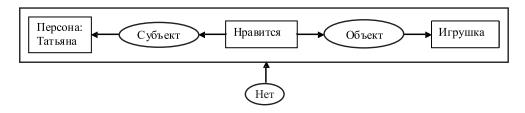


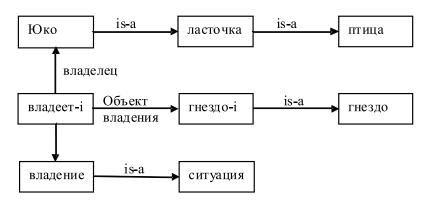
Рис. 8.17 – Логическое отрицание.

Используя отрицание и конъюнкцию, можно представить на концептуальных графах дизъюнкцию. Однако для упрощения схем ее представляют в виде специального бинарного отношения «или», аргументами которого являются вершиныпредложения.

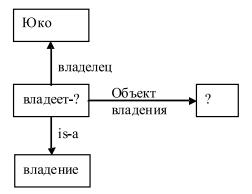
По своим выразительным возможностям концептуальные графы эквивалентны логике исчисления предикатов. Существует возможность взаимного перехода

от описаний предметной области на языке исчисления предикатов к концептуальным графам, и наоборот. Выбор тех или иных средств определяется характером решаемых задач и склонностями разработчиков СИИ.

Методы вывода на семантических сетях. Особенность семантической сети (в то же время ее недостаток) заключается в целостности системы, выполненной на ее основе, не позволяющей разделить базу знаний и механизм выводов. Обычно интерпретация семантической сети определяется с помощью использующих ее процедур. Эти процедуры основаны на нескольких способах, но наиболее типичный из них—это способ сопоставления частей сетевой структуры. Он основан на построении подсети, соответствующей вопросу, и сопоставлении ее с базой данных сети. При этом для исчерпывающего сопоставления с базой данных вершинам переменных подсети присваиваются гипотетические значения. Рассмотрим в качестве примера следующий процесс сопоставления с частью базы данных. Когда в отношении следующей части базы данных [12, 22]



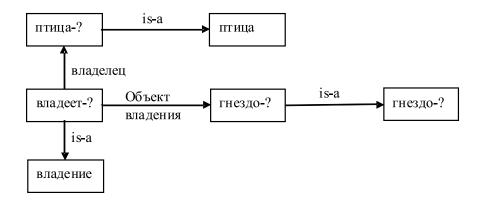
возникает вопрос «Чем владеет Юко?», он представляется в виде подсети (общая форма):



и проводится сопоставление. При этом сначала отыскивается вершина «владеть», имеющая ветвь «владелец», направленную к вершине «Юко», затем следует соединение с вершиной, которая показывает ветвь «собственность», и ответ на вопрос.

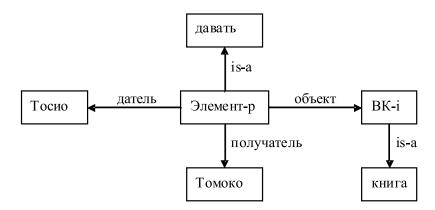
Посмотрим теперь, как выполняется вывод. Например, вопрос «Существует ли птица, которая владеет гнездом?» можно представить с помощью подсети следующим образом:

Однако эта подсеть не годится для непосредственного сопоставления с базой данных. Для этого из узла «Юко» к узлу «птица» проводится дуга is_a (означающая, что Юко является птицей). Теперь становится возможным соединение



«владеть-?» с «владеет-і» и «гнездо-?» с «гнездо-і» и проведение сопоставления. В результате ответ на вопрос звучит как «Да, — это Юко».

Рассмотрим теперь вкратце способ вывода, именуемый перекрестным поиском. Этот способ означает поиск отношений между концептуальными объектами и ответ на вопрос путем обнаружения узла, в котором пересекаются дуги, идущие от двух различных узлов. Например, на вопрос «какие отношения между Тосио и Томоко» для следующей подсети



можно получить ответ «Тосио отдал эту книгу (ВК-і) Томоко».

Очевидно, излишне напоминать о том, что в семантических отношениях узлов и дуг семантической сети не должно быть противоречий [8].

Систематизация отношений конкретной семантической сети зависит от специфики знаний предметной области и является сложной задачей. Особого внимания заслуживают общезначимые отношения, присутствующие во многих предметных областях. Именно на таких отношениях основана концепция семантической сети. В семантических сетях, так же как при фреймовом представлении знаний, декларативные и процедурные знания не разделены, следовательно, база знаний не отделена от механизма вывода. Процедура логического вывода обычно представляет совокупность процедур обработки сети. Семантические сети получили широкое применение в системах распознавания речи и экспертных системах.

8.4.3 Фреймы

История фреймов. Если брать более ранние времен, американский психолог Ф. Бартлетт в книге «Память» [23] обнаружил, что при вербализации прошлого

опыта люди регулярно пользуются стереотипными представлениями о действительности. Такие стереотипные фоновые знания Бартлетт называл схемами. Например, схема квартиры включает знания о кухне, ванной, прихожей, окнах и т. п. Характерная для России схема поездки на дачу может включать такие компоненты, как прибытие на вокзал, покупка билета на электричку и т. д. Наличие схематических представлений, разделяемых языковым сообществом, решающим образом влияет на форму порождаемого дискурса. Это явление было заново «открыто» в 1970-е годы, когда появился целый ряд альтернативных, но весьма близких по смыслу терминов. Так, американские специалисты в области искусственного интеллекта предложили термины «фрейм» [17] и «скрипт» [25]. «Фрейм» в большей степени относится к статическим структурам (типа модели квартиры), а «скрипт» — динамическим (типа поездки на дачу или посещения ресторана), хотя сам Минский предлагал использовать термин «фрейм» и для динамических стереотипных структур.

Следует заметить, что еще до Минского термин «фрейм», а также производные «фрейминг» и «рейфрейминг» использовались Э. Гоффманом и его последователями в социологии и социальной психологии для обозначения различных способов видения общественно значимых проблем (ядерная энергетика может подводиться под фреймы «прогресс», «взбесившаяся технология», «сделка с дьяволом»), а также средств, используемых для поддержания того или иного видения. Особое значение эти термины имеют также в прикладной коммуникативно-психологической методике, известной как нейролингвистическое программирование (НЛП).

Не следует забывать и о роли американского лингвиста Ч. Филлмора, который к нуждам лингвистики адаптировал понятие «фрейм». В его статьях начала 1980-х годов была предложена концепция фреймовой семантики, представляющая собой адаптацию для нужд лексической семантики понятия «фрейм» — структуры для представления знаний, предложенной в искусственном интеллекте Минским. Использование понятия фрейма позволяет описывать, каким образом понимание языковых выражений определяется человеческими знаниями о тех ситуациях, которые этими выражениями описываются: так, мы по-разному понимаем относительное расположение сидящих в выражениях «кот на стене» и «муха на стене», обозначаемое предлогом «на». Синтезом идей падежной грамматики, фреймовой семантики и взглядов на роль в языке фразеологических формул стала концепция конструкционной грамматики, разрабатываемая и преподаваемая Филлмором в 1990-х годах совместно с П. Кеем. Конструкционная грамматика постулирует в качестве главной единицы языка комплексную грамматическую конструкцию, своего рода грамматический фрейм, в котором представлена одновременно лексическая, грамматическая, синтаксическая и семантическая информация.

В 1990-х годы Филлмор заинтересовался идеями корпусной лингвистики и совместно с Д. Журавским возглавил долгосрочную (1997–2003) программу FrameNet, направленную на создание большого компьютерного словаря лексических единиц и грамматических конструкций во фреймовом представлении и последующего обеспечения к нему доступа в онлайновом режиме.

Основные понятия концепции фреймов. В книге [7] П. Джексон предполагается, что представление понятий в мозге не требует строгого формулирования набора свойств, которыми должна обладать та или иная сущность, чтобы можно

было рассматривать ее в качестве представителя определенной категории сущностей. Многие из тех категорий, которыми мы пользуемся, не имеют четкого определения, а базируются на довольно расплывчатых понятиях. Создается впечатление, что человек больше всего внимания обращает на те бросающиеся в глаза свойства, которые ассоциируются с объектами, наиболее ярко представляющими свой класс.

Такие объекты были названы прототипами.

В системе фреймов предпринимается попытка судить о классе объектов, используя представление знаний о прототипах, которые хорошо представляют большинство разновидностей объектов данного класса, но параметры этого фрейма не определены полностью, и их предстоит еще выявить.

А. П. Частиков [21] выделяет фреймы-образцы (прототипы), хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных. Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

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

Иная классификация используется в книге Л.А. Растригина [20]. Когда все параметры определены, то фрейм называется фреймом-примером, обобщением многих фактов является фрейм-прототип.

Согласно Дж. Ф. Люгеру [14] ячейки фрейма содержат следующую информацию:

- Данные для идентификации фрейма.
- Взаимосвязь этого фрейма с другими фреймами. «Гостиничный телефон» может служить специальным экземпляром «телефона», который может служить примером «механизма связи».
- Дескрипторы требований для фрейма. Стул имеет высоту сидения от 20 до 40 см, его задняя часть выше 60 см. и т. д. Эти требования могут быть использованы для определения соответствия новых объектов стереотипу фрейма.
- Процедурная информация об использовании данной структуры. Важной особенностью фреймов является возможность присоединить к ячейке процедурный код.
- Информация по умолчанию. Это значения ячейки, которые должны быть истинными, если не найдены противоположные. Например, стулья имеют 4 ножки, на телефонах есть кнопки, гостиничные кровати заправляются персоналом.
- Информация для нового экземпляра. Многие ячейки фрейма могут оставаться незаполненными, пока не указано значение для отдельного экземпляра или пока они не понадобятся для некоторого аспекта решения задачи. Например, цвет кровати может оставаться неопределенным.

Пример фрейма. Возьмем описание гостиничного номера из книги Люгера. Сам номер и его компоненты могут быть описаны рядом отдельных фреймов. Помимо кровати во фрейме должен быть представлен стул: ожидаемая высота — от 20 до 40 см, число ножек — 4, значение по умолчанию — предназначен для сидения. Далее фрейм представляет гостиничный телефон: вариант обычного телефона, но расчет за переговоры связывается с оплатой гостиничного номера. По умолчанию существует специальный гостиничный коммутатор, и человек может использовать этот телефон для заказа еды в номер, внешних звонков и получения других услуг. На рис. 8.18 изображен фрейм, представляющий гостиничный номер.

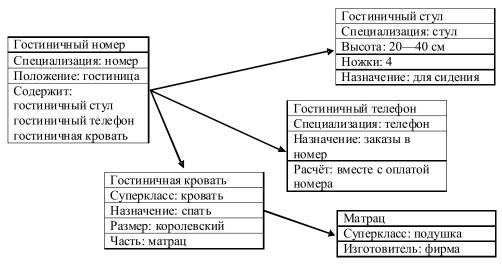


Рис. 8.18

Каждый фрейм можно рассматривать как семантическую сеть, состоящую из выделенных вершин и связей. Верхний уровень фрейма представляет соответствующее понятие, а последующие уровни—терминальные слоты, которые содержат конкретные значения.

Прототипом фрейма в семантической сети является подсеть, состоящая из некоторой сущности и характеризующих ее свойств (события, его признаков и атрибутов). Например, если под событием понимать ПОЕЗД №1, отправляющийся из Санкт-Петербурга в 23 часа 55 минут и прибывающий в Москву в 7 часов 55 минут, соответствующая ему подсеть представляется графом, представленным на рис. 8.19.

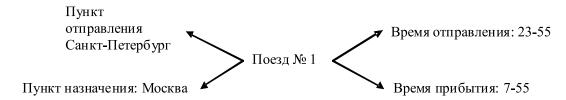


Рис. 8.19 – Подсеть неоднородной семантической сети.

Поезд №1 считается истинным событием, если имеют место приведенные значения атрибутов, т. е. он обладает соответствующими свойствами. Как реляционная структура данных этот граф представляется в табличной форме (табл. 8.2).

Номер	Пункт отправле-	Пункт назначе-	Время от-	Время прибы-
поезда	ния	ния	правления	ТИЯ
1	Санкт-	Москва	23–55	7–55
	Петербург			

Таблица 8.2

При представлении таблицы в виде фрейма поля записи называются его слотами. Таблица содержит имена и значения слотов, а ее название структурируется как имя фрейма.

Модель фрейма была предложена М. Минским с целью борьбы с высокой размерностью сетевых моделей. Каждый фрейм можно рассматривать как фрагмент сети, объединяющий сущности на основе семантической близости. Например, с помощью фрейма можно описать такую локальную ситуацию, как обстановка в комнате, при составлении сетевой модели квартиры.

Принципиальными особенностями фреймовой модели, отличающими ее от реляционной модели данных, являются:

- возможность смешанного заполнения слотов константами и переменными;
- возможность наличия пустых слотов;
- размещение в слотах указателей на другие фреймы (наследование частей) для создания сети;
- размещение в слотах имен выполняемых процедур.

Использование аппарата ссылок позволяет реализовывать сложные модели представления знаний путем детализации любых понятий фреймов. Например, могут потребоваться знания о других поездах. Тогда в слот вместо поезда №1 будет помещена ссылка на список поездов. Там могут быть помещены сведения об уровне сервиса, количестве свободных мест и т. д. В свою очередь, если потребуется пояснить характеристики свободных мест (купе, плацкарт, верхнее, нижнее), по ссылке заводится фрейм, содержащий эти характеристики. Аналогичным образом детализируется уровень сервиса. Таким образом, с помощью аппарата ссылок формируется иерархическая фреймовая сеть. Дочерний фрейм, на который делается ссылка, находится в отношении Part Of к своему родителю.

Наличие процедур в слотах означает, что фреймовая модель является смешанной моделью представления знаний, в основу которой положена декларативная составляющая.

Фреймом-прототипом является фрейм, у которого значения слотов являются переменными, а фреймом-экземпляром — фрейм, у которого все слоты являются константами. Фреймы-экземпляры находятся в отношении Is A к фреймупрототипу.

Операции над фреймами. Над фреймами можно совершать некоторые теоретико-множественные операции, например объединение и пересечение. При объединении в результирующем фрейме будут присутствовать все слоты, встречающиеся в исходных фреймах. В слотах, не являющихся общими, остаются прежние значения, слоты с одинаковыми именами объединяют свои значения. При пересечении фреймов в результирующем фрейме присутствуют только общие для

исходных фреймов слоты. Их значения вычисляются двумя способами. Либо в результирующем фрейме присутствуют общие для исходных фреймов значения, либо значения исходных фреймов также подвергаются пересечению. Фреймовые системы подразделяются на статические и динамические, последние допускают изменение фреймов в процессе решения задачи.

Наследование в теории фреймов. А. П. Частиков [21] выделяет важнейшим свойством теории фреймов так называемое наследование свойств. Во фреймах наследование происходит по AKO (A-Kind-Of-9то) связям. Слот AKO указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т. е. переносятся, значения аналогичных слотов. Наследование может быть частичным, когда часть свойств не наследуется.

П. Джексон [7] описывает также и множественное наследование. В настоящее время является общепризнанным, что некоторый фрейм может наследовать информацию от множества предшественников в системе фреймов. В результате граф, представляющий связи между фреймами, стал больше походить на решетку, чем на дерево, поскольку каждый узел не обязательно имеет единственного предшественника. Очень часто система строится таким образом, что некоторые фреймы имеют несколько предшественников, хотя в подавляющем большинстве структур сохраняется единственность корня.

Такая организация связей между фреймами не влечет за собой никаких проблем до тех пор, пока информация, поступающая из различных источников наследования, не становится противоречивой.

Механизмы вывода на фреймах. В интеллектуальных системах с фреймовым представлением знаний используется три способа управления логическим выводом: демоны, присоединенные процедуры и механизм наследования, который является и единственным для фреймовых объектно-ориентированных систем.

Управленческие функции механизма наследования заключаются в автоматическом поиске и определении значений слотов фреймов нижележащих уровней по значениям слотов фреймов верхних уровней, а также в запуске присоединенных процедур и демонов.

Основной механизм вывода из фреймовой модели — сопоставление (matching). Различают синтаксическое сопоставление, когда сравнивается структура единиц знания (фреймов или слотов), и семантическое сопоставление, когда сравнивается содержимое этих единиц. Результат сопоставления может быть бинарным или иметь параметрический характер (введенный параметр отражает степень сопоставимости образцов).

Существует ряд языков представления знаний, основанных на фреймовой модели. В них присоединенные процедуры реализуются на ЛИСПе. К таким языкам относятся FRL, KRL и др. Фреймовую модель можно создавать и на обычных языках программирования, например C++.

Универсализм фреймовой модели приводит к такому разнообразию конкретных реализаций, что достоинства и недостатки определяются уже не фреймовой идеологией, а конкретной реализацией. Характерными чертами фреймовых языков являются:

• Представление иерархической модели понятий ПО и отвечающей ей совокупности экземпляров.

- Реализация связей и закономерностей ПО присоединенными процедурами.
- Семантическое сопоставление понятий при поиске по образцу.

В настоящее время универсальным средством для представления знаний являются объектно-ориентированные языки программирования, которые способны реализовать все особенности фреймовой модели.

Основным преимуществом фрейма как способа представления знаний является наглядность и гибкость в употреблении.

Кроме того, фреймовая структура согласуется с современными представлениями о хранении информации в памяти человека.

Основной недостаток фреймовых систем — отсутствие специального механизма управления выводом. Вывод обычно реализуется с помощью процедур, встроенных во фреймы.

8.4.4 Сценарии

История сценариев. Согласно экспериментальным данным люди организуют знания в структуры, соответствующие типовым ситуациям. Читая статью о ресторане, бейсболе или политике, мы устраняем любые двусмысленности в тексте с учетом тематики статьи. Если сюжет статьи неожиданно меняется, человек делает краткую паузу в процессе чтения, необходимую для модификации структуры знания. Возможно, слабоструктурированный текст тяжело понять именно потому, что мы не можем с легкостью связать его с какой-нибудь из структур знания. В этом случае ошибки понимания возникают из-за того, что мы не можем решить, какой из контекстов использовать для разрешения неопределенных местоимений и других двусмысленностей разговора [14].

Сценарии первоначально предложили Schank и Abelson [25]. Фрейм в большей степени относится к статическим структурам, а скрипт — к динамическим. Очень часто никакого различия между понятиями «скрипт» и «сценарий» не проводится, при этом обычно в русском языке используется второй термин.

Основные понятия сценариев.



Сценарий — это структурированное представление, описывающее стереотипную последовательность в частом контексте [14].

.....

Сценарии используются в системах понимания естественного языка для организации базы знаний в терминах ситуаций, которые система должна понимать.

Сценарий включает следующие компоненты:

- 1) Начальные условия, которые должны быть истинными при вызове сценария. В этом сценарии они включают открытый ресторан и голодного посетителя, имеющего некоторую сумму денег.
- 2) Результаты или факты, которые являются истинными, когда сценарий завершается; например клиент сыт и его деньги потрачены, владелец ресторана имеет больше денег.

- 3) Предположения, которые поддерживают контекст сценария. Они могут включать столы, официантов и меню. Множество предположений описывает принятые по умолчанию условия реализации сценария. Например, предполагается, что ресторан должен иметь стулья и столы, если не указано противоположное.
- 4) Роли являются действиями, которые совершают отдельные участники. Официант принимает заказы, доставляет пищу и выставляет счет. Клиент делает заказ, ест и платит.
- 5) Сцены. Шенк разбивает сценарий на последовательность сцен, каждая из которых представляет временные аспекты сценария. Сценарий похода в ресторан включает в себя следующие сцены: вход, заказ, принятие пищи и т. д.

Элементы сценария — основные «части» семантического значения — представляются отношением концептуальной зависимости. Собранные вместе во фреймоподобной структуре они представляют последовательность значений или событий.

Пример сценария. Как пример возьмем сценарий посещения ресторана, представленный в книге [14].

Большинство людей в ресторане полностью спокойны (т. е. они знают, чего ожидать и как действовать). Их либо встречают на входе, либо знак указывает, куда они должны пройти, чтобы сесть. Меню либо лежит на столе, либо предоставляется официантом, либо посетитель его просит. Мы понимаем процедуры для заказа еды, принятия пищи, оплаты счета и ухода (рис. 8.20).

В действительности ресторанный сценарий совершенно отличается от других сценариев принятия пищи, например в ресторане быстрого обслуживания или в рамках официально семейного обеда. В сети быстрого обслуживания заказчик входит, становится в очередь, оплачивает заказ до принятия пищи, ожидает подноса с едой, берет поднос, пытается найти свободный стул и т. д. Это — две различные последовательности событий, и каждая имеет свой сценарий.

Недостатки сценариев. Дж. Ф. Люгер описывает две проблемы сценариев и способы их решения.

Пакеты организации памяти решают проблему негибкости сценария, представляя знания с помощью более мелких компонентов вместе с правилами их динамического связывания в схемы, соответствующие текущей ситуации [25]. Организация памяти знаний особенно важна для реализации рассуждений, основанных на опыте, в которых решатель должен эффективно находить в памяти подходящее решение прежней проблемы.

Проблемы организации и поиска знаний очень трудны и характерны для моделирования семантического значения. Евгений Черняк показал, какое количество знаний требуется для понимания даже простых детских рассказов. Рассмотрим фрагмент текста, описывающий вечеринку в честь дня рождения: «В день рождения Мэри подарили два воздушных змея, поэтому она вернула одного из них в магазин». Мы должны знать о традиции дарить подарки в день рождения; знать, что такое воздушный змей и почему Мэри могла не захотеть иметь двух змеев. Мы должны знать о магазинах и их политике обмена. Несмотря на эти проблемы, программы, использующие сценарии и другие семантические представления, могут понимать естественный язык в ограниченных предметных областях. Приме-

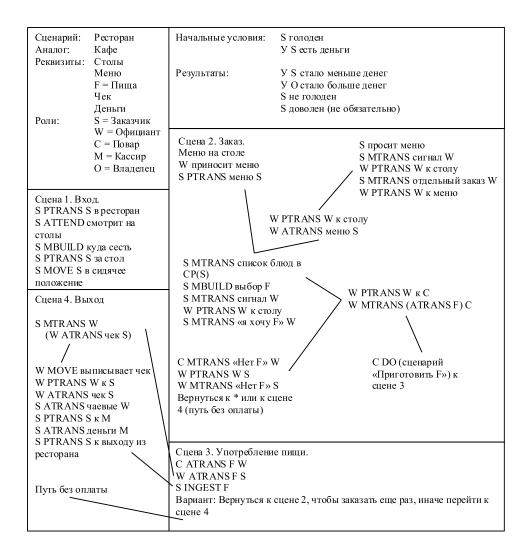


Рис. 8.20 – Сценарий посещения ресторана.

ром может служить программа, которая интерпретирует сообщения, поступающие по каналам новостей. Благодаря использованию сценариев природных катастроф, успешных сделок и других стереотипных историй, программа достигла заметных успехов в этой ограниченной, но реальной области [Schank и Riesback, 1981].



Контрольные вопросы и задания к главе 8

......

- 1) Определите понятие «знание».
- 2) Опишите модуль предметной области.
- 3) Чем отличаются понятия «знания» и «модель»?
- 4) Поясните смысл понятий «представление» и «представление знаний».
- 5) Приведите пример универсального языка представления знаний.
- 6) Какие функции может выполнять система представления знаний?
- 7) Сформулируйте понятия «данные».
- 8) Опишите свойства знаний.
- 9) Какие школы могут использоваться для отношений между информационными единицами.
- 10) Сравните классическую технологию разработки программного обеспечения и технологию разработки ИнС.
- 11) Приведите обобщенную схему представления данных и знаний.
- 12) Назовите три основные модели данных.
- 13) Проанализируйте экстенциональные и интенсиональные представления в базах данных и интеллектуальных системах.
- 14) Перечислите и кратко опишите модели представления знаний.
- 15) Поясните смысл понятий «предметная область», «проблемная область».
- 16) Дайте определение понятию «сущность» и рассмотрите категории (классы) сущностей.
- 17) Опишите знаковое представление понятий.
- 18) Что представляет выражение треугольника фреи?
- 19) Охарактеризуйте схему и формулу понятий.
- 20) Поясните экстенсионал и интенсионал понятия.
- 21) Постройте связи между категориями, используемые для описания понятий.
- 22) Перечислите основные типы абстрагирования понятий.
- 23) Дайте характеристику понятию «формальная система».
- 24) Определите понятие алфавита и формулы.
- 25) Какие выражения можно построить из символов алфавита?
- 26) Определите различие между ППФ и аксиомой.
- 27) Что вы назовете правилом логического вывода?
- 28) Расскажите о двух классах формальных систем: исчисление высказываний и исчисление предикатов.

- 29) Представьте правила вывода в исчислении предикатов.
- 30) Дайте определение семантической сети.
- 31) Охарактеризуйте типы отношений в семантической сети.
- 32) Классифицируйте семантические сети по типам отношений между понятиями.
- 33) С помощью каких функций в книге Андрейчикова А. В. семантические сети Куиллиена систематизируют функции отношений между понятиями?
- 34) Разработайте семантическую сеть, показывающую взаимоотношения птицы и самолета.
- 35) Что понимает Гаскаров Д. В. под совокупностью сущностей, понятий и ситуаций предметной области?
- 36) Охарактеризуйте понятие «процедура» в семантической сети.
- 37) Проанализируйте нелексические свойства семантических сетей.
- 38) Приведите классификацию методов и моделей обобщения знаний.
- 39) Приведите примеры обобщенных, индивидных и агрегатных объектов.
- 40) Приведите и поясните типы связей между объектами семантической сети.
- 41) Рассмотрите примеры семантических сетей с разными типами связей.
- 42) Как Куиллиан предложил определять функции между понятиями в семантической сети?
- 43) Покажите на примере способы описания семантических сетей: блочными структурами и концептуальными графами.
- 44) Проанализируйте методы вывода на семантических сетях.
- 45) Перечислите фамилии ученых-исследователей, имеющих отношение к истории и развитию фреймовых систем.
- 46) Основные понятия концепции фреймов П. Джексона, А. П. Частикова, Ф. Люгера.
- 47) Какие теоретико-множественные операции можно совершать над фреймами?
- 48) Опишите свойство расследования в фреймах.
- 49) Назовите три способа управления логическим выводом в фреймах.
- 50) Определите основные преимущества и недостатки фреймов.
- 51) Сформулируйте определение понятия «сценарий».
- 52) Какие компоненты включает сценарий?
- 53) Приведите пример сценария.

.....



Литература к главе 8

- [1] Попов Э. В. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта / Э. В. Попов, Г. Р. Фридман. М. : Наука, 1976. 456 с.
- [2] Андрейчиков А. В. Интеллектуальные информационные системы : учебник / А. В. Андрейчиков, О. Н. Андрейчикова. М. : Финансы и статистика, 2006.-424 с.
- [3] Вагин В. Н. Дедукция и обобщение в системах принятия решений / В. Н. Вагин. М.: Наука, 1988. 384 с.
- [4] Гаврилова Т. А. Базы знаний и интеллектуальных систем : учебник / Т. А. Гаврилова, В. Ф. Хорошевский. СПб: Изд-во «Питер», 2000. С. 19–23.
- [5] Гаскаров Д. В. Интеллектуальные информационные системы : учеб. для вузов / Д. В. Гаскаров. М. : Высш. шк., 2003. 431 с.
- [6] Горский Д. П. Вопросы абстракции и образования понятий / Д. П. Горский. М.: Изд-во АН СССР, 1961.
- [7] Джексон П. Введение в экспертные системы / П. Джексон. М. : Вильямс, 2001.-624 с.
- [8] Змитрович А. И. Интеллектуальные информационные системы / А. И. Змитрович. Минск: HTOOO «ТетраСистемс», 1997. 368 с.
- [9] Искусственный интеллект : в 3 кн. / под ред. Э. В. Попова. М. : Радио и связь, 1990. Кн. 1: Системы общения и экспертные системы. 464 с.
- [10] Бондарев В. Н. Искусственный интеллект : учеб. пособие для вузов / В. Н. Бондарев, Ф. Г. Аде. Севастополь : Изд-во СевНГУ, 2002. 615 с.
- [11] Логический подход к искусственному интеллекту / под ред. Г. П. Гаврилова. — М. : Мир, 1990.
- [12] Лорьер Ж.-Л. Системы искусственного интеллекта : пер. с франц. / Ж.- Л. Лорьер. М. : Мир, 1991.-568 с.
- [13] Луценко Е. В. Интеллектуальные информационные системы / Е. В. Луценко. Краснодар: КубГАУ, 2006. 615 с.
- [14] Люгер Дж. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем / Дж. Ф. Люгер. М. : Вильямс, 2005. 864 с.

- [15] Макетирование, проектирование и реализация диалоговых информационных систем / под ред. Е. И. Ломако [и др.]. М. : Финансы и статистика, 1993.-320 с.
- [16] Микони С. В. Модели и базы знаний : учеб. пособие / С. В. Микони. СПб. : СПГУПС, 2000. 155 с.
- [17] Минский М. Фреймы для представления знаний / М. Минский. М. : Энергия, 1979.
- [18] Представление и исследование знаний : пер. с япон. / X. Уэно, M. Исидзука. M. : Мир, 1989. 220 с.
- [19] Павлов С. Н. Интеллектуальные информационные системы : учеб. пособие / С. Н. Павлов. Томск: Томский межвузовский центр дистанционного образования, 2004.-328 с.
- [20] Растригин Л. А. С компьютером наедине / Л. А. Растригин. М. : Радио и связь, 1990.-224 с.
- [21] Частиков А. П. Разработка экспертных систем. Среда CLIPS / А. П. Частиков, Т. А. Гаврилова, Д. Л. Белов. СПб. : БХВ-Петербург, 2003.-606 с.
- [22] Эндрю А. Искусственный интеллект / А. Эндрю. М.: Мир, 1985.
- [23] Bartlett F. Remembering / F. Bartlett. Cambridge: Cambridge University Press, 1932. P. 38–46.
- [24] Fillmore C. J. The case for case. In Bach and Harms / C. J. Fillmore, 1968. P. 1–88.
- [25] Шенк, Абельсон (Schank R., Abelson R). Scripts Plans and Knowledge, Advance Papers of Fourth Intern. Joint Conf. on Artif. Inttel., 1975, №2, р. 151–157 [Русский перевод: Труды IV Межд. конф. по искусств. интеллекту, т. 6. М.: Научн. совет по компл. пробл. «Кибирнетика» АН СССР, 1975, с. 208–220].

Глава 9

СИСТЕМЫ ПОНИМАНИЯ ЕСТЕСТВЕННОГО ЯЗЫКА, МАШИННЫЙ ПЕРЕВОД

Для понимания текста интеллектуальная система должна отвечать на все прямые вопросы, а также представлять ситуации реального мира по тексту. Она должна включать три уровня интерпретации: морфологический анализ и предсинтез, синтаксический анализ, семантический анализ.

Подсистемы языковых форм общения интеллектуальных систем должны понимать предложения ЕЯ, выводить ответ на основании имеющихся у систем фактов, выражать ответ на естественном языке, накапливать и корректировать свои знания.

Любая развитая система искусственного интеллекта должна иметь интеллектуальный интерфейс. Эта проблема тесно связана с проблемой автоматического перевода с одного языка на другой.

Машинный перевод — это перевод, строящийся на использовании машиной определенных и постоянных для данного вида материала соответствий между словами и грамматическими явлениями разных языков.

Выделяют два подхода к разработке систем машинного перевода: традиционный (основанный на правилах) и статистический (основанный на статистической обработке словарных баз).

Системы с непрямым переводом делятся на системы типа «transfer» и типа «interlingua». Алгоритмы перевода для систем типа «transfer» строятся как композиция трех процессов: анализ входного предложения в терминах структур входного языка, преобразование этой структуры в аналогичную структуру выходного языка и затем синтез выходного предложения по полученной структуре.

Системы типа «interlingua» предполагают наличие некоторого промежуточного языка, на котором можно описать все структуры как входного, так и выходного языков.

9.1 Искусственный интеллект и компьютерная лингвистика

Под термином «компьютерная лингвистика» обычно понимается широкая область использования компьютерных инструментов — программ, компьютерных технологий организации и обработки данных — для моделирования функционирования языка в тех или иных условиях, ситуациях, проблемных сферах, а также сфера применения компьютерных моделей языка в лингвистике и смежных дисциплинах [2].

В узком смысле проблематика компьютерной лингвистики часто связывается с моделированием общения человека с ЭВМ на естественном языке (для этого создаются специальные системы обработки естественного языка), а также с теорией и практикой информационно-поисковых систем. К компетенции компьютерной лингвистики относится машинный перевод.

Язык можно рассматривать с двух точек зрения:

- средство организации, представления, передачи, усвоения и развития знаний когнитивный феномен;
- универсальный механизм общения людей (либо непосредственно, либо через компьютер) коммуникативный феномен.

Основными проблемными целями компьютерной лингвистики являются:

- понимание компьютером глубинного содержания больших массивов естественных текстов;
- повышение уровня естественности диалога между человеком и компьютером (режима меню и свободного общения на естественном языке (ЕЯ));
- создание больших банков данных и знаний на основе лексических единиц естественного языка;
- создание многоязычных информационных систем;
- общение с компьютером в режиме звучащей речи.

Представим тезаурус основных направлений развития работ в области компьютерной лингвистики, но не фундаментальных, а практических для различных прикладных систем.

- 1) Системы обработки письменного текста на естественном языке.
- 2) Системы диалога пользователя с компьютером на естественном языке.
- 3) Системы машинного перевода с одного естественного языка на другой естественный язык.
- 4) Системы наполнения баз знаний и баз данных на основе естественно- языковых источников.
- 5) Системы вывода знаний на основе «логики» естественного языка.
- 6) Системы автоматического построения текстов на естественном языке.
- 7) Системы интеллектуальных машинных словарей.
- 8) Системы общения в режиме звучания речи.

9.2 Понимание текстов на естественном языке

Любая развитая система искусственного интеллекта должна иметь интеллектуальный интерфейс, т.е. систему естественно-языкового общения с ЭВМ. Эта проблема тесно связана с проблемой автоматического перевода с одного языка на другой.

Основой решения этих проблем являются два класса логико-лингвистических моделей — анализирующие и синтезирующие. В первом случае речь идет об извлечении смысла из текста, во втором — о кодировании высказывания на внутреннем языке ЭВМ средствами естественного языка. Они используются в автоматическом переводе, информационном поиске, автоматическом реферировании, в диалоге с ЭВМ на ограниченном естественном языке или при создании интеллектуального интерфейса.

Системы ЕЯ общения и автоматического перевода имеют общие проблемы:

- формирование словарных фондов;
- автоматический анализ и синтез на уровнях морфологии, синтаксиса и семантики;
- описание дискура (общей организации текста).

При восприятии текста сначала формируется его промежуточное когнитивное представление. Это представление является не конечным результатом, а лишь средством, позволяющим соотнести информацию, содержащуюся в сообщении, с системой знаний, на основе чего происходит ее понимание. Далее формируется целостное представление о содержании текста как о фрагменте действительности, описываемой этим текстом (рис. 9.1).



Рис. 9.1 – Схема понимания текста на естественном языке.

Функции понимания текста:

- ответы на вопросы по тексту;
- пополнение текста знаниями из базы;
- представление ситуаций реального мира.

9.2.1 Уровни понимания текста интеллектуальной системы

Введем компоненты уровня понимания текста [4]:

- Т текст на естественном языке, представленный для ввода в систему;
- E расширенный текст, включающий условия его порождения в некоторой среде;
- P расширенный текст, включающий все невербальные элементы, относящиеся к субъекту, передающему текст;

TR — правила пополнения текста, описывающего его структуру;

ER — правила пополнения текста, опирающиеся на знания о коммуникации и психологии личности;

А — ответ, формулируемый системой;

К — знания, хранящиеся в БЗ;

F — факты, хранящиеся в базе фактов;

FRK — правила порождения знаний из фактов;

КМ — метафорические знания;

KRM — правила порождения знаний метафорического уровня.

<u>Первый</u> уровень характеризуется схемой $T \to A$, т. е. ответ системой формируется только на основе прямого содержания (рис. 9.2).

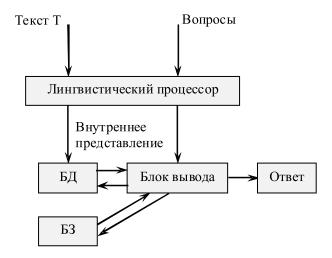


Рис. 9.2 – Схема первого уровня понимания.

Лингвистический процессор обеспечивает перевод вербального представления текста и вопросов во внутреннее их представление. Блок вывода обеспечивает описание предметной области и выводы на знаниях.

<u>Второй</u> уровень характеризуется схемой $(T,TR) \to A$ или $(T,TR,ER) \to A$, где добавляется средство логического вывода из информации, содержащейся в Т. Это специальные логики отношений (псевдофизические логики) времени, причины, пространства, действий, которые позволяют решать задачи получения новых знаний с необходимой полнотой и имитацией нестрогих рассуждений человека. Например, логика времени для интеллектуальной системы включает формальную систему, позволяющую выводить временные отношения, неявно присутствующие в описании ситуаций (на основании в памяти устройства базовых отношений и правил вывода), и систему декларативных описаний (в виде семантических сетей, фреймов, сценариев), с помощью которых восстанавливаются отношения, явно присутствующие в описании ситуации (рис. 9.3).

Эти уровни практически реализованы во многих интеллектуальных системах.

<u>Третий</u> уровень характеризуется схемой $(E, TR, ER) \to A$, где используется расширенный текст E, получаемый при наличии двух каналов—источников информации (например, текст и зрение), что характерно для интеллектуальных роботов. Здесь осуществляется соотношение текста с реальной ситуацией его порождения.

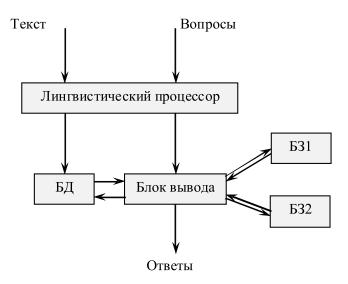


Рис. 9.3 – Схема второго уровня понимания.

<u>Четвертый</u> уровень характеризуется схемой $(P, TR, PR) \rightarrow A$. На этом уровне развитие получают разнообразные способы вывода на знаниях и данных. Здесь используются модели индивидуального и группового поведения.

На следующих уровнях (их два) возникают новые процедуры для манипулирования знаниями, но уже на более высоком уровне. Их называют метауровни понимания текста.

9.2.2 Уровни интерпретации понимания текста

Для понимания текста система должна отвечать на все прямые вопросы, а также представлять ситуации реального мира по тексту. Она должна включать три уровня интерпретации.

- 1) Морфологический анализ и предсинтез, который заключается в приписывании каждой словоформе (отрезки текста между пробелами) морфологической информации.
- 2) Синтаксический анализ, осуществляющий грамматический разбор предложений, это построение синтаксической структуры в виде дерева или даже графа.
- 3) Семантический анализ—это построение семантического графа предложения для выявления смысла предложения.

Суть понимания текста или анализа заключаются в том, что сначала текст разбивается на словоформы — отрезки текста между пробелами (с участием знаков препинания).

Каждой словоформе приписывается морфологическая информация. Существуют два способа морфологического анализа: декларативный и процедуральный. При декларативном способе словарь содержит все возможные словоформы каждой основы, с приписываемой морфологической информацией. Составление такого словаря со всеми словоформами для каждой основы представляет собой трудоемкую операцию и требует большого объема памяти. Тогда морфологический анализ сво-

дится к поиску словоформы с приписываемой ей морфологической информацией. При процедурном способе словарная зона системы состоит из словаря основ и словаря аффиксов. Причем их (основ и аффиксов) значительно меньше, чем словоформ. Тогда морфологический анализ сводится к алгоритмам приписывания каждой словоформе морфологической информации.

Например, представим морфологический анализ словоформам «машины» и «имеются» в предложении «Какие вычислительные машины имеются в Томских институтах СОАН?»

Тип — неодушевленное существительное

машины Род — женский

Число — множественное, единственное Падеж — именительный, родительный

Тип — глагол

имеются Залог — действительный

Время — настоящее

Число — множественное

Далее, при синтаксическом анализе предложений устанавливается соответствие между строками морфологической информации или между их цепочками. Грамматический разбор предложения это выделение подлежащего, сказуемого, дополнения и т. д., между которыми устанавливаются связи в виде дерева зависимостей. Представим синтаксический анализ для нашего предложения (рис. 9.4).

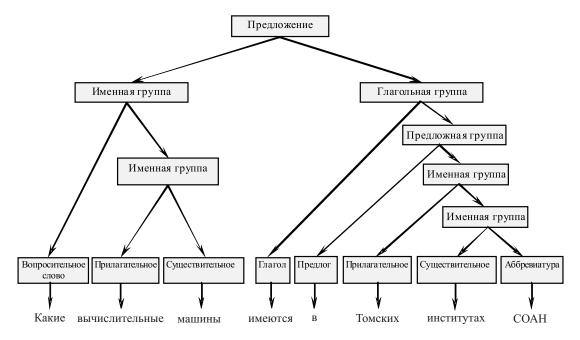


Рис. 9.4 - Синтаксический анализ предложения

Синтаксический анализ, ограниченный или полный, более сложен, чем морфологический. В нем в настоящее время применяют теорию трансформационных грамматик (ТГ) или трансформационных преобразований языковых выражений, когда сохраняются семантические инварианты. В системах, основанных на

ТГ, каждому предложению входного ТЕЯ ставятся в соответствие две различные структуры: глубинная и поверхностная.

Для понимания текста необходим и семантический анализ. На этом этапе происходит выявление смысла предложения в виде семантического графа. Для этого выявляются слова, которые можно представить в виде фреймов, определяющих смысл предложения. В рассматриваемом предложении такими словами-фреймами будет предикат «имеются» со слотами <что?>, <где?>, <у кого?> и слово «институты» с единственным слотом «какие?».

Семантическое представление имеет вид графа. Значительная часть моделей семантического анализа относятся к моделям типа «Смысл — текст». Эта модель рассматривает семантическое представление связного фрагмента текста как состоящего из двух компонентов: семантического графа и сведений о коммуникативной организации смысла.

Современные системы понимания ТЕЯ, как правило, ориентированы на предметную область и в качестве средств понимания используют лингвистические знания в виде словарей, грамматик и знаний о предметной области в виде структуры семантической сети, в которой фиксированы типовые ситуации.

Семантический граф предложения покажем для нашего примера на рис. 9.5.

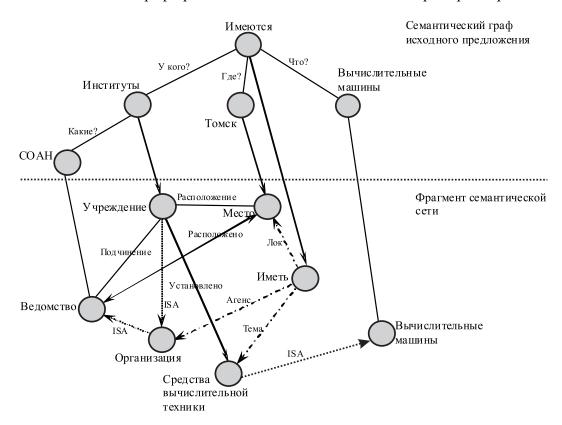


Рис. 9.5 - Семантический анализ предложения.

В верхней части рисунка показан семантический граф предложения, а в нижней — фрагмент семантической сети предметной области. Во фрагменте семантической сети выделены вершины-понятия: «ведомство», «организация», «учреждение», «место», «иметь», «средства вычислительной техники» и «вычислительные машины», а также показаны отношения между этими понятиями в виде дуг: «под-

чинение» и «расположение». Они характеризуют предметную область. Отношение ISA (есть — некоторые) — родо-видовое, «агенс», «лок», «тема» — отношения лингвистической природы. Эти отношения характерны для глагола «имеется».

Наложение семантического графа на сеть позволяет осуществить семантическую интерпретацию запроса и в конечном итоге сформировать запрос на языке внутреннего представления. Далее, обращение к базе данных (экстенсиональной семантической сети) и позволит сформулировать ответ во внутреннем представлении.

Понимание ТЕЯ на синтаксическом уровне. В процессе анализа содержания текста иногда делают попытки учесть: 1) не только лингвистические, но и логикосемантические отношения между языковыми объектами; 2) контекст, определяющий лингвистические отношения, который выходит за пределы дискура и текста; 3) не только сведения о данной ПО, но и связи с внешним миром в целом.

Таким образом, в семантическом анализе осуществляется глобальное понимание ТЕЯ.

Синтез связных текстов. Проблемы синтеза связного текста возникают в системах машинного перевода, реферирования текстов, диалоговых системах, экспертных системах и заключаются в представлении (передаче) взаимосвязанных цепочек предложений на естественном языке из внутреннего содержания (памяти) ЭВМ. Синтез текста состоит из трех этапов:

- определение содержания текста и последовательное развертывание его во времени;
- запись будущего текста на внутреннем языке системы;
- перевод текста на ЕЯ.

Любой текст предполагает адресата, читателя или слушателя. Это отражается на общей организации текста, которая исследуется теорией дискура. Она отражает законы построения устных и письменных текстов и строится на базе следующих принципов.

- 1) Форма, в которой информация появляется в тексте, не обязательно совпадает с формой, в которой она хранится в памяти человека (в БЗ).
- 2) Наряду с общечеловеческими законами построения текстов имеются законы, характерные для носителей данного языка, для данной группы людей.
- 3) При построении текстов необходимо учитывать их жанровые особенности. Можно выделить четыре схемы дискура:
- схема текста с общим атрибутивным значением;
- схема с общим значением идентификации;
- схема с общим значением состава;
- схема сравнения и противопоставления.

Выбор схемы дискура определяется генератором текста в зависимости от цели («доказать», «сравнить», «описать», «объяснить» и т. д.).

9.2.3 Структура языковых форм общения интеллектуального робота

Опишем структуру и общие принципы построения подсистемы языковых форм общения интеллектуальных систем. В этом случае система должна решать по крайней мере следующие задачи.

- 1) Понимать предложения естественного языка (возможно, ограниченного).
- 2) Выводить ответ на основании имеющихся у систем фактов.
- 3) Выражать ответ на естественном языке.
- 4) Накапливать и корректировать свои знания.

Например, представим структуру подсистемы языковых форм общения интеллектуального робота по основным этапам вопросно-ответной системы [7] (рис. 9.6).

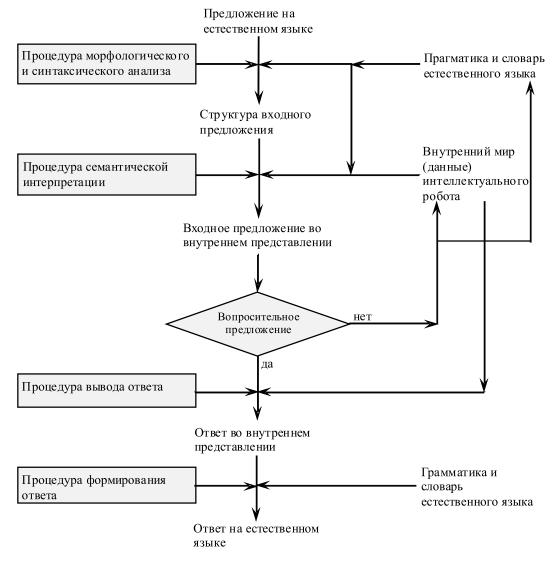


Рис. 9.6 – Структура подсистемы языковых форм общения интеллектуального робота.

9.3 Машинный перевод

В настоящее время имеется достаточно широкий выбор пакетов программ, облегчающих труд переводчика, которые условно можно подразделить на две основные группы: электронные словари (electronic dictionary) и системы машинного перевода (machine translation system) [3].

9.3.1 Электронные словари

Электронные словари начали разрабатываться значительно раньше, чем альтернативные им системы МП. Они предоставляют пользователю возможность быстрого поиска лишь одного отдельного слова (термина) или фразеологизма.

В настоящее время на рынке программного обеспечения имеется чрезвычайно широкий выбор словарей — от самых простейших (например, DIC) до мощных систем, объединяющих в одной программной оболочке несколько лексических баз данных, — специализированных тематических словарей, последовательность подключения которых определяется пользователем (LINGVO). Выбор того или иного программного продукта зависит исключительно от аппаратных возможностей компьютера, которым располагает пользователь, финансовых соображений и конкретных условий, в которых работает переводчик.

Потенциальную аудиторию этих программ составляют учащиеся и студенты, использующие словарь как одно из пособий при изучении иностранного языка, а также домашние и бизнес-пользователи, которым нужен универсальный и простой в применении словарь-справочник для эпизодического перевода деловых бумаг и корреспонденции [9].

9.3.2 Понятие машинного перевода

Датой рождения машинного перевода (как области исследований) принято считать март 1947 г., когда Уоррен Уивер (Warren Weaver), директор отделения естественных наук Фонда Рокфеллера, написал Норберту Винеру письмо, в котором сравнивал задачу перевода с задачей дешифровки текста.



Машинный перевод (МП) — выполняемое на компьютере действие по преобразованию текста на одном естественном (исходном) языке в эквивалентный по содержанию текст на другом (целевом) языке, а также результат такого действия [6].

Выделяют следующие формы организации взаимодействия ЭВМ и человека при машинном переводе:

- с постредактированием: исходный текст перерабатывается машиной, а человек-редактор исправляет результат;
- с предредактированием: человек приспосабливает текст к обработке машиной (заменяет в исходном тексте сложные для перевода конструкции более

простыми, раскрывает сокращения, приводит текст к формату, исключающему неоднозначное восприятие), после чего начинается программная обработка;

- с интерредактированием: человек участвует в процессе перевода, разрешая трудные случаи;
- смешанные системы (например, одновременно с пред- и постредактированием).

Коммуникативная эквивалентность нового текста по отношению к оригиналу обеспечивается выполнением трех основных требований:

- текст перевода должен в возможно более полном объеме передавать содержание оригинала, что, прежде всего, означает недопустимость произвольного опущения или добавления информации;
- текст перевода должен соответствовать нормам языка перевода, так как их нарушение, по меньшей мере, создает помехи для восприятия информации, а иногда ведет и к ее искажению;
- текст перевода должен быть примерно сопоставим с оригиналом по своему объему, чем обеспечивается сходство стилистического эффекта с точки зрения лаконичности или развернутости выражения.

Круг деятельности, охватываемой понятием «перевод», очень широк. Переводятся с одного языка на другой стихи, художественная проза, научные и научнопопулярные книги из различных областей знания, дипломатические документы, деловые бумаги, статьи политических деятелей и речи ораторов, газетные материалы, беседы лиц, разговаривающих на разных языках и вынужденных прибегать к помощи посредника — «толмача», дублируются кинофильмы.

9.3.3 Семантические проблемы машинного перевода

Различия в категоризации. Известно, что стандартное членение суток на временные отрезки, зафиксированное в русском языке, отличается от членения суток, принятого в немецком и английском языках. Если в русском языке сутки членятся на четыре отрезка (утро, день, вечер, ночь), то, например, в немецком представлены шесть временных отрезков: Morgen 'yrpo', Vormittag 'предполуденное время', Miltag 'полдень', Nachmillag 'послеполуденное время', Abend 'вечер', Nachl 'ночь'. Таким образом, русское предложение Встретимся сегодня вечером может быть в принципе переведено на немецкий язык и как Wir treffen uns heute nachmillag «Встретимся сегодня в послеполуденное время» и как Wir treffen uns heute abend «Встретимся сегодня вечером», так что только обращение к контексту позволяет выбрать адекватный вариант перевода. Членение суток, представленное в английском языке, отличается как от русского, так и от немецкого. В теоретическом смысле эта проблематика активно разрабатывалась в трудах представителей европейского неогумбольдтианства (Л. Вайсгербер и др.), а также американских ученых, сформулировавших теорию лингвистической относительности (Э. Сепир, Б. Уорф), и их последователей, отстаивавших тезис о неповторимом своеобразии зафиксированной в каждом языке «картины мира» [11].

Грамматические различия. Среди межъязыковых различий, обусловленных несовпадениями в организации языковой структуры, значимыми для теории перевода могут оказаться в первую очередь грамматические различия. Р. О. Якобсон подчеркивает, что грамматическая структура (в отличие от структуры лексикона) определяет те значения, которые обязательно должны быть выражены в данном языке.

В общем виде проблему передачи грамматических категорий можно представить следующем образом: пусть в языке L_1 есть некая грамматическая категория, а в языке L_2 ее нет, и наоборот. Например, в английском языке есть грамматическая категория определенности-неопределенности, выражаемая артиклями, а в русском нет. Означает ли это, что этой категории всегда должен быть найден некоторый, не обязательно грамматический, аналог при переводе с английского на русский? Это разумно делать только в том случае, если значение грамматической категории оказывается коммуникативно значимым. Например, словосочетание *on the table* в зависимости от ситуации может переводиться и как *'на столе'*, и как *'на этом столе'*.

В обратном случае, например при переводе с безартиклевого языка на языки, использующие артикли, при отсутствии коммуникативной значимости постановка артиклей регулируется стандартными грамматическими правилами, которые могут отличаться от языка к языку. Например, русское предложение Oh cmydehm переводится на немецкий язык как Er ist Student (без артикля вообще), а на английский He is a student (с неопределенным артиклем).

Грамматические различия как фактор метафоризации. Чисто грамматические категории могут «лексикализироваться», т. е. осмысляться как содержательно значимые. Так, Якобсон писал о том, что И. Е. Репина в свое время удивил тот факт, что немецкие художники изображают грех в виде женщины (немецкое слово die Siinde 'грех' — женского рода). Смерть в русской культуре — это старуха с косой, в немецкой же — der Sensenmann , скелет с косой, поскольку русское слово смерть — это существительное женского рода, в то время как немецкое der Tod 'смерть' — мужского рода.

Наиболее очевидна потенциальная значимость подобных различий в переводе художественных текстов, о чем писал академик Л. В. Щерба в своем известном анализе лермонтовского перевода *Сосны* Г. Гейне: <...> совершенно очевидно <...>, что мужской род *(Fichlenbaum, а не Fichte)* не случаен и что в своем противопоставлении женскому роду *Palme* он создает образ мужской неудовлетворенной любви к далекой, а потому недоступной женщине. Лермонтов женским родом сосны отнял у образа всю его любовную устремленность и превратил сильную мужскую любовь в прекраснодушные мечты».

Скрытые категории. Помимо случаев межъязыковых расхождений в грамматикализации «классических» грамматических категорий наподобие определенности-неопределенности, встречаются случаи, в которых существенные различия не грамматикализуются систематическим образом, что делает их очень сложными для систематического описания. Такие категории часто называются «скрытыми» (англ. covert; термин принадлежит Б. Уорфу) или «имплицитными». Для русского языка примером скрытой категории может служить категория отчуждаемой-неотчуждаемой принадлежности. В русском языке различие между отчуждаемой и неотчуждаемой

принадлежностью хотя и не систематически, как в случае открытых грамматических категорий, но все же проводится в большей степени по сравнению с западноевропейскими языками. Например, по-немецки (как и в большинстве других германских языков) русским словосочетаниям велосипедная цепь и цепь от велосипеда соответствует одна и та же агглютинативная конструкция Fahrradkette. Иными словами, немецкий язык не делает разницы между отчужденной (цепь от велосипеда, валяющаяся, скажем, на свалке) и неотчужденной принадлежностью (велосипедная цепь как деталь велосипеда). Выбор правильного перевода слов типа Fahrradkette на русский язык зачастую оказывается непростой задачей.

Важную информацию о возможностях перевода выражений с категорией неотчуждаемой принадлежности можно получить из знаний об устройстве мира. Если некоторая сущность является частью другой сущности, то следует иметь в виду, что для нее обсуждаемая категория может оказаться существенной.

Перевод идиом и пословиц. При переводе идиом, пословиц и поговорок наличие эквивалентной единицы в языке L_2 само по себе далеко не всегда обеспечивает адекватный результат. Возможность понимания дословного перевода идиоматического выражения языка L_1 человеком, родным для которого является язык L_2 , объясняется наличием в L_2 каких-то других выражений, опирающихся на понятийные структуры, сопоставимые с понятийными структурами, стоящими за соответствующим языковым выражением L_1 .

В случае, когда переводчик не учитывает отсутствия когнитивных параллелей между L_1 и L_2 и переводит буквально идиому языка L_1 на L_2 , у читателя возникают серьезные проблемы с пониманием текста. Сравним дословный перевод русской идиомы в гробу видать из примера (1) на немецкий язык (2).

- $(1) < \ldots >$ я ш не говорю тебе там про Кандинского шмандинского или про Клее там шмее так што ш ты гонишь мне опять про своих Пикассо шмикассо Утрилло шмутрилло, я ш в гробу видел твоего Ван Гога там шмангога $< \ldots >$ (В. Сорокин. «Дорожное происшествие».)
- (2) <...> ich rede ja auch nicht iiber Kandindkij Schmandinski oder iiber Klee dings Schmee also was quatschst du mich wieder voll mit deinem Picasso Schmikasso Utrillo Schmulrillo, ich hab ja deinen Gogh dings Schmangoch <...> im Sarg gesehen (V. Sorokin. Voriall auf der Strafie).

Практически ни один носитель немецкого языка не в состоянии понять, что в (2) имеется в виду.

На этом примере становится очевидной значимость культурной составляющей, заложенной во внутренней форме некоторых идиом.

9.3.4 Процесс машинного перевода



Процесс машинного перевода представляет собой последовательность преобразований, применяемых к входному тексту и превращающих его в текст на выходном языке, который должен максимально воссоздавать и, как правило, структуру исходного текста, но уже средствами выходного языка [5].

.....

Говоря о машинном переводе, следует, прежде всего, помнить, что компьютер—создание бездушное. Мышления как такового при машинном переводе не происходит: предложение расчленяется на части речи, в нем выделяются стандартные конструкции, слова и словосочетания переводятся по находящимся в памяти машины словарям. Затем переведенные части речи собираются по правилам другого языка [6].

В первых системах машинного перевода исходный текст постепенно превращался в текст на выходном языке путем замены всех его элементов, найденных в словаре на переводные эквиваленты (прямой, пословный перевод). Для систем данного типа характерны бинарность, отсутствие промежуточных структур (анализ ведется сразу в категориях выходного языка), одновариантность. Их сильная сторона — простота устройства и большая скорость работы [5].

Морфологическое представление входного предложения непосредственно перерабатывалось в выходной текст. Однако очень скоро выявилась недостаточность морфологического анализа, и в поисках дальнейших путей развития выделились системы с непрямым переводом, которые имеют два направления—системы типа transfer и типа interlingua.

В системах типа transfer после морфологического анализа следует синтаксический анализ, в ходе которого выстраивается синтаксическая структура входного предложения в виде размеченного дерева. В качестве узлов дерева выступают слова или словосочетания, а связи между ними отражают синтаксические связи внутри предложения.

При таком переводе каждое предложение проходит последовательность преобразований, состоящих из трех частей (этапов):

АНАЛИЗ→ТРАНСФЕР (межъязыковые операции)→СИНТЕЗ.

В свою очередь, каждый из этих этапов представляет собой достаточно сложную систему промежуточных преобразований.

Цель этапа АНАЛИЗА — построить структурное описание (промежуточное представление, внутреннее представление) входного предложения.

В практике создания систем машинного перевода различают следующие уровни анализа:

- *досинтаксический анализ* (морфологический анализ, анализ оборотов, неопознанных элементов текста);
- синтаксический анализ (строит синтаксическое представление предложения);
- *семантический анализ* (строит аргументно-предикатную структуру высказываний или другой вид семантического представления предложения и текста);
- концептуальный анализ (анализ в терминах концептуальных структур, отражающих семантику программного обеспечения).

Задача этапа ТРАНСФЕРА (собственно перевода) — преобразовать структуру входного предложения во внутреннюю структуру выходного предложения. К этому этапу относятся и замены лексем входного языка их переводными эквивалентами (лексические межьязыковые преобразования). Компонент трансфера задается двумя частями:

- 1) Словарь трансфера, или переводной (межъязыковой) словарь, это двуязычный словарь лексических замен.
- 2) Словарь структурных соответствий, или грамматика трансфера.

Цель этапа СИНТЕЗА— на основе полученной в результате анализа структуры построить правильное предложение выходного языка. Одна из задач синтеза—линеаризация узлов дерева, определяющая нужный порядок слов в синтезируемом предложении. Синтез проходит те же уровни, что и анализ, но в обратном направлении.

Процесс перевода заключается в следующем. Сначала выполняется анализ предложения в терминах лингвистических структур входного языка, затем построенная структура преобразуется в аналогичную структуру для выходного языка, то есть на основе дерева входной фразы строится дерево выходной. И, наконец, по полученной структуре производится синтез выходного предложения, который подразделяется на синтаксический синтез (перевод синтаксического дерева фразы в цепочке морфологических представлений слов) и морфологический синтез (создание нужных форм из основ и аффиксов).

Идея альтернативного подхода была предложена в 1949 году Уореном Уивером. Она состоит в том, чтобы использовать для перевода промежуточный язык — «universal language», или «interlingua». У этой идеи есть два больших преимущества и один принципиальный недостаток. Главное преимущество состоит в радикальном сокращении числа пар языков, между которыми нужно выполнять перевод. Кроме того, универсальный язык — посредник — как нельзя лучше подходит для хранения документов, требующих многоязыкового представления, а если он еще будет читаемым для человека, то некоторые документы можно сразу создавать на промежуточном языке.

К сожалению, универсальный язык пока не создан. Причина заключается в чрезвычайно высоких требованиях к нему, а именно:

- 1) он должен иметь достаточно выразительных возможностей, чтобы охватить все, что можно высказать на языках, для которых он служит посредником;
- 2) должна быть возможность переводить на него и обратно без каких-либо потерь или искажений.

Попытки создать системы типа interlingua периодически повторяются, но пока не дали ни одной работающей системы.

9.3.5 Системы машинного перевода

Стремительные потоки информационного обмена между высокоразвитыми промышленными странами, лавина научно-технической документации, поступающая от производителей товаров и современных технологий, требуют совершенно нового подхода к проблеме перевода технической литературы. Выход один: максимально автоматизировать процесс, оставив человеку его творческую редакционную часть. В этом помогает система машинного перевода.

В отличие от словаря программа не делает подстрочный перевод каждого слова в отдельности, а выполняет именно связный перевод с учетом грамматических,

синтаксических и других особенностей языка. Существуют различные технологии разработки систем машинного перевода, однако суть машинного перевода состоит именно в выполнении программой связного перевода текста на основе анализа особенностей входного и выходного языков.

На сегодняшний день широко известны два основных подхода к разработке систем машинного перевода: традиционный (основанный на правилах, rule-based machine translation) и статистический (основанный на статистической обработке словарных баз, statistical-based machine translation).

Традиционный машинный перевод

Технология, которая называется rule-based machine translation (то есть основанная на обработке лингвистических алгоритмов) — традиционный подход, на сегодняшний день применяется наиболее широко. Ее использует большинство разработчиков: SYSTRAN (Франция), Linguatec (Германия) и российская компания ПРОМТ.

Для осуществления машинного перевода такой программой в компьютер вводится специальная программа, реализующая алгоритм перевода, под которым понимается последовательность однозначно и строго определенных действий над текстом для нахождения переводных соответствий в данной паре языков L_1-L_2 при заданном направлении перевода (с одного конкретного языка на другой). Система машинного перевода включает в себя двуязычные словари, снабженные необходимой грамматической информацией (морфологической, синтаксической и семантической). Они предназначены для обеспечения передачи эквивалентных, вариантных и трансформационных переводных соответствий. Кроме того, система машинного перевода включает алгоритмические средства грамматического анализа, реализующие какую-либо из принятых для автоматической переработки текста формальных грамматик. Имеются также отдельные системы машинного перевода, рассчитанные на перевод в рамках трех и более языков.

Существуют разные вариации алгоритмов перевода. И в каждом конкретном случае последовательность операций определяется разработчиком.

В общем случае, работа систем машинного перевода состоит из следующих этапов.

- 1) Сначала система осуществляет морфологический анализ слов (указывает род, число, лицо и другие морфологические характеристики для каждого слова). Кроме того, программа фиксирует полную информацию по многозначным словам (тем словам, которые могут относиться к разным частям речи или иметь разные значения).
- 2) Затем происходит объединение отдельных слов в группы: именные (где главным словом является существительное, а зависимые от него слова определяются по идентичным морфологическим характеристикам), глагольные (главное слово глагол) и др. Кроме того, в этот момент система может решить вопрос многозначности для некоторых слов в зависимости от их контекста.
- 3) На следующем этапе программа приступает к определению членов предложения и их роли в предложении, границ и типа связи между простыми

предложениями. Сначала она ищет границы простых предложений, которые определены знаками препинания. Затем определяет главные члены, причем сначала система ищет сказуемое и только потом подлежащее перед ним (если перед сказуемым подлежащего нет, программа ищет его за сказуемым или делает вывод, что подлежащее отсутствует (например, в безличных предложениях)). Завершив поиск главных членов предложения, система определяет сферы их влияния (слова и группы слов, зависимые от подлежащего и от сказуемого). Все группы, которые система не смогла отнести ни к сфере влияния подлежащего, ни к группе сказуемого, считаются обстоятельствами.

4) На заключительной стадии работы происходит окончательное согласование всех членов предложения и построение предложений с учетом требований грамматики выходного языка. Элементы согласуются внутри групп, а также уточняется их зависимость от подлежащего или сказуемого и подтверждается порядок слов в предложении.

Наиболее распространенной является следующая последовательность операций процесса машинного перевода:

- 1) На первом этапе осуществляется ввод текста и поиск входных словоформ (слов в конкретной грамматической форме, например дательного падежа множественного числа) во входном словаре (словаре языка, с которого производится перевод) с сопутствующим морфологическим анализом, в ходе которого устанавливается принадлежность данной словоформы к определенной лексеме (слову как единице словаря). В процессе анализа из формы слова могут быть получены также сведения, относящиеся к другим уровням организации языковой системы.
- 2) Следующий этап включает в себя:
 - а) перевод идиоматических словосочетаний, фразеологических единств или штампов данной предметной области (например, при англо-русском переводе обороты типа in case of, in accordance with получают единый цифровой эквивалент и исключаются из дальнейшего грамматического анализа);
 - б) определение основных грамматических (морфологических, синтаксических, семантических и лексических) характеристик элементов входного текста (например, числа существительных, времени глагола, синтаксических функций словоформ в данном тексте и пр.), производимое в рамках входного языка.
 - в) разрешение омографии (конверсионной омонимии словоформ скажем, англ. round может быть существительным, прилагательным, наречием, глаголом или же предлогом); лексический анализ и перевод лексем. Обычно, на этом этапе, однозначные слова отделяются от многозначных (имеющих более одного переводного эквивалента в выходном языке). После чего однозначные слова переводятся по спискам эквивалентов, а для перевода многозначных слов используются так называемые контекстологические словари, словарные статьи которых

представляют собой алгоритмы запроса к контексту на наличие/отсутствие контекстных определителей значения.

- 3) Окончательный грамматический анализ, в ходе которого доопределяется необходимая грамматическая информация с учетом данных выходного языка (например, при русских существительных типа *сани*, *ножницы* глагол должен стоять в форме множественного числа, при том что в оригинале может быть и единственное число).
- 4) Синтез выходных словоформ и предложения в целом на выходном языке. В зависимости от особенностей морфологии, синтаксиса и семантики конкретной языковой пары, а также направления перевода общий алгоритм перевода может включать и другие этапы, а также модификации названных этапов или порядка их следования, но вариации такого рода в современных системах, как правило, незначительны. Анализ и синтез могут производиться как пофразно, так и для всего текста, введенного в память компьютера; в последнем случае алгоритм перевода предусматривает определение так называемых анафорических связей (такова, например, связь местоимения с замещаемым им существительным скажем, местоимения им со словом местоимения в самом этом пояснении в скобках) [10].

Таким образом, процесс перевода системы логически понятен: происходит поиск языковых эквивалентов, их объединение по морфологическим признакам, синтаксический анализ членов предложения и окончательный синтез предложения на выходном языке.

Система проводит большую работу, формируя связный перевод предложения на основе информации о каждом слове и согласуя между собой все члены предложения. В данном случае важную роль играют полнота словаря, ориентированность его на содержание и набор языковых средств переводимых текстов, эффективность способов разрешения лексической многозначности, результативность работы алгоритмов извлечения грамматической информации, нахождения переводных соответствий и алгоритмов синтеза.

Статистический машинный перевод

В последнее десятилетие наметилась тенденция к использованию систем машинного перевода, основанных на статистическом анализе [8].

Данная технология перевода получила широкое распространение благодаря разработкам компании Google. Поисковый портал Google представил пользователям онлайновый сервис с ВЕТА-версиями некоторых языковых направлений, разработанных на основе статистической технологии.

Статистический машинный перевод—это разновидность машинного перевода текста, основанная на сравнении больших объемов языковых пар. Языковые пары—тексты, содержащие предложения на одном языке и соответствующие им предложения на втором. Чем больше в распоряжении имеется языковых пар и чем точнее они соответствуют друг другу, тем лучше результат статистического машинного перевода.

В процессе работы система анализирует огромные словарные базы парных фрагментов (фраз из двух-трех слов) — оригинал фрагмента и его перевод.

Программа вычисляет наиболее вероятную последовательность слов выходного языка, которую она считает соответствующей переводу исходного текста. В отличие от традиционных систем перевода статистическая программа не учитывает в своей работе грамматические правила, что, безусловно, отражается на качестве перевода. Нередки случаи, когда вместо связного перевода статистический переводчик выдает совершенно несогласованные предложения, лишенные всякого смысла.

Возможность быстрого пополнения словарных баз, безусловно, является преимуществом статистического метода машинного перевода. Однако у этого обстоятельства есть и обратная сторона: большое количество информации требует немалого объема памяти для хранения. Возможно, в будущем, эту проблему удастся успешно решить, но на данный момент она существует.

Существует мнение, что статистический машинный перевод более перспективен и на выходе предлагает более качественный перевод, чем традиционная технология. Но пока практика доказывает обратное. Одна из причин низкого качества перевода статистической системы — недостаточный объем словарных баз. Яркие примеры перевода (отрывки новостей информационного агентства Reuters) с английского языка на русский представлены на рисунке 9.7.

ПРИМЕР	МАШИННЫЙ ПЕРЕВОД НА ОСНОВЕ ПРАВИЛ (PROMT)	СТАТИСТИЧЕСКИЙ МАШИННЫЙ ПЕРЕВОД (Google)			
Venus Williams powered into the Japan Open semi- finals with a 7—6 6—	Уимблдонская чемпионка Венера Williams приводила в действие в Японию Открытые полуфиналы с 7—6 6—3 победы над Алисией Molik Австралии в четверг	Чемпион Уимблдона Венеры Уильямс энергии в Японии Открыть полуфабрикатов финала с 3 июня 6 июля завоевать Австралии Алисия Молик в			
Pachaurl, an Indian scientist, said that rising temperatures this century could bring risks for the extinction of up to 30 percent of the world's species. A creeping rise in sea levels could threaten Pacific Islands and many coastlines	сказал, сказал, что возрастающие температуры в этом столетии могли принести риски для исчезновения до 30 процентов разновидностей в мире. Ползающее	могли бы риски для исчезновению до 30 % мирового видов. Ползучее повышение уровня моря может угрожать островов Тихого океана и многих			

Рис. 9.7 – Примеры перевода разными методами машинного перевода.

Переводы были выполнены с помощью системы PROMT (традиционный машинный перевод) и онлайн-портала Google (статистический машинный перевод).

Из приведенных примеров видно, что на данный момент сервис Google не всегда справляется с переводом имен и названий, возникают сложности с согласованием членов предложения, и получившийся вариант перевода больше напоминает

подстрочник. Вполне вероятно, что в будущем разработчикам удастся повысить качество переводов, выполняемых этим методом машинного перевода, однако есть опасение, что такой системе не хватает применения лингвистических алгоритмов, анализа особенностей входного и выходного языков.

Машинный перевод, основанный на правилах, также имеет погрешности и на данный момент не может предложить идеального качества перевода. Однако в целом перевод, основанный на применении правил, выглядит более гладко, и смысл текста понятен сразу.

Современные системы машинного перевода укомплектованы большим количеством полезных инструментов для повышения качества перевода. В частности, можно зарезервировать (сохранить) названия организаций или имена, и тогда программа оставит их непереведенными, что позволит избежать нелепостей при переводе.

Кроме того, можно подключать к системе словари, содержащие специализированную лексику. Такие словари обычно предлагают сами разработчики программ.

Есть еще одна полезная возможность: пользователь может самостоятельно создавать собственные словари и заносить в них именно те варианты перевода, которые ему необходимы. Подключение пользовательского словаря также позволит корректно перевести необходимую информацию. В современных системах перевода можно задавать правила перевода, предлагать на выбор несколько вариантов перевода одного слова и сохранять заданные настройки в виде шаблонов, чтобы не задавать их каждый раз заново.

Освоение полезных возможностей программы помогает на выходе получить достойное качество перевода и существенно сократить время на его редактирование. Важно учитывать и то, что с помощью системы перевода можно всего за несколько секунд получить доступ к необходимой информации (получить черновой вариант перевода), то есть сэкономить время и быстро обработать большой объем текста, что сегодня актуально как никогда.

9.3.6 Достоинства систем машинного перевода

Всем хорошо известно, что хороший перевод текста—это не только творческая, но и достаточно трудоемкая работа. Что касается творческой части, то в обозримом будущем в соревновании компьютер-человек всегда победит «живой» переводчик. Однако для решения проблем, обусловленных трудоемкостью процесса перевода, системы машинного перевода могут оказаться хорошим подспорьем. Для того чтобы это лучше понять, перечислим достоинства систем машинного перевода:

- 1) Высокая скорость. Всего несколько секунд, и пользователь получает перевод многостраничного текста. Это позволяет быстро понять смысл текста, а если система настроена на перевод текстов этой тематики, требуется минимальная редакторская правка.
- 2) Низкая стоимость. Это очень легко оценить: при обращении к профессиональным переводчикам, приходится платить за каждую страницу переведенного текста достаточную сумму, а если предприятие нанимает штатного переводчика, то ему приходится платить зарплату. В случае с системой машинного перевода нужно платить деньги только один раз при покупке

- программы. Принятие решения: чему отдать предпочтение, остается за человеком.
- 3) Доступ к услуге. Немаловажный фактор, который многие критики систем машинного перевода не принимают в расчет. Программа-переводчик всегда под рукой, а обращаться в переводческое бюро во многих случаях связано с дополнительными затратами времени и сил.
- 4) Конфиденциальность. Системе машинного перевода можно доверить любую информацию.
- 5) Универсальность. Любой переводчик всегда имеет специализацию, т. е. переводит тексты по той теме, которой он хорошо владеет. Система машинного перевода выгодно отличается тем, что она абсолютно универсальна. Нужно только грамотно подключить специализированный словарь по соответствующей тематике.
- 6) Пополнение специализированных словарей системы новейшими терминами значительно опережает аналогичные словари полиграфического исполнения. В ряде случаев также рекомендуется вести свой собственный словарь новых терминов или новых значений.
- 7) Возможность перевода информации в Интернете. В on-line режиме наиболее ярко проявляются все преимущества систем машинного перевода. Более того, в большинстве случаев переводить информацию в Интернете можно только с помощью программ-переводчиков. Именно эта потребность обусловила огромный рост интереса к системам машинного перевода сейчас в мире. Только благодаря on-line системам машинного перевода появилась возможность просматривать иностранные сайты, не затрудняясь с их переводом. Кроме того, здесь действуют все вышеперечисленные достоинства систем машинного перевода: перевод текстов по любой тематике производится быстро и конфиденциально.
- 8) Единообразие стиля и используемой терминологии. Как известно, затраты на постредактирование при работе коллектива переводчиков составляют около 100–140 % от стоимости перевода. Перевод, выданный системой машинного перевода, гораздо легче править, поскольку он выдержан в одном стиле. Если в тексте, какой-либо часто встречающийся термин переведен неправильно, то все эти ошибки можно исправить простой автозаменой. Когда объемный текст переводится группой переводчиков, то приходится вылавливать отдельные неточности, допущенные каждым переводчиком. Редактору в этом случае требуется также «выравнивать» и стиль перевода.
- 9) Отсутствие затрат на форматирование. Это особенно важно при переводе электронной документации. Программа-переводчик полностью сохраняет исходное форматирование, что позволяет сэкономить время и деньги при подготовке перевода.

9.3.7 Будущее машинного перевода

Можно выделить два основных стимула к развитию работ по машинному переводу в современном мире.

Первый — собственно научный; он определяется комплексностью и сложностью компьютерного моделирования перевода. Как вид языковой деятельности перевод затрагивает все уровни языка — от распознавания до передачи смысла высказывания и текста. Кроме того, для перевода характерна обратная связь и возможность сразу проверить теоретическую гипотезу об устройстве тех или иных языковых уровней и эффективности предлагаемых алгоритмов. Эта характеристическая черта перевода вообще и машинного перевода в частности привлекает внимание теоретиков, в результате чего продолжают возникать все новые теории автоматизации перевода и формализации языковых данных и процессов.

Второй стимул — социальный, и обусловлен он возрастающей ролью самой практики перевода в современном мире как необходимого условия обеспечения межъязыковой коммуникации, объем которой возрастает с каждым годом. Другие способы преодоления языковых барьеров на пути коммуникации — разработка или принятие единого языка, а также изучение иностранных языков — не могут сравниться с переводом по эффективности. С этой точки зрения можно утверждать, что альтернативы переводу нет, так что разработка качественных и высокопроизводительных систем машинного перевода способствует разрешению важнейших социально-коммуникативных задач.

Перспективы развития машинного перевода связаны с дальнейшей разработкой и углублением теории и практики перевода, как машинного, так и «человеческого». Для развития теории важны результаты сопоставительного языкознания, общей теории перевода, теории закономерных соответствий, способов представления знаний, оптимизации и совершенствования лингвистических алгоритмов. Новые и более эффективные словари с необходимой словарной информацией, строгие теории терминологизации лексики, теория и практика работы с подъязыками помогут повысить качество перевода лексических единиц. Формальные грамматики, ориентированные на перевод, дадут возможность оптимизировать алгоритмы нахождения переводных соответствий в данной коммуникативной ситуации, которая может быть описана в рамках соответствующих прикладных теорий представления знаний. Наконец, новые возможности программирования и вычислительной техники также будут вносить свой вклад в совершенствование и дальнейшее развитие теории и практики машинного перевода.



Контрольные вопросы и задания к главе 9

.....

......

- 1) Что такое компьютерная лингвистика? Назовите ее проблемные цели.
- 2) Перечислите проблемы систем ЕЯ общения и автоматического перевода.
- 3) Как происходит понимание текстов на естественном языке?
- 4) Дайте краткую характеристику уровней понимания текста.
- 5) Назовите уровни интерпретации понимания текста. Дайте их краткое описание.
- 6) В чем состоит суть анализа текста?
- 7) Что происходит на уровне синтаксического анализа текста?
- 8) Назовите этапы синтеза текста?
- 9) Опишите структуру подсистемы языковых форм общения интеллектуального робота.
- 10) Что такое машинный перевод?
- 11) Какие существуют формы организации взаимодействия ЭВМ и человека при машинном переводе?
- 12) Какими требованиями обеспечивается коммуникативная эквивалентность?
- 13) Перечислите семантические проблемы машинного перевода.
- 14) Назовите направления систем с непрямым переводом. Опишите процесс перевода для каждого направления.
- 15) Чем отличается система машинного перевода от электронного словаря?
- 16) Расскажите о подходах к разработке систем машинного перевода.
- 17) Дайте характеристику основным этапам процесса машинного перевода.
- 18) Как можно повысить качество перевода при статистическом подходе?
- 19) Назовите достоинства систем машинного перевода.
- 20) Какие системы машинного перевода Вам известны?



Литература к главе 9

[1] Архипов А. Ф. Самоучитель перевода с немецкого языка на русский / А. Ф. Архипов. — М. : Высш. шк., 1991 - 255 с.

.....

- [2] Баранов А. Н. Введение в прикладную лингвистику / А. Н. Баранов. 2-е изд. М. : УРСС, 2003. 360 с.
- [3] Вейзе А. А. Перевод технической литературы с английского языка на русский / А. А. Вейзе, Н. Б. Киреев, К. Мирончиков. Минск, 1997. 111 с.
- [4] Гаскаров Д. В. Интеллектуальные информационные системы: учеб. для вузов / Д. В. Гаскаров. М. : Высш. шк., 2003. 431 с.
- [5] Искусственный интеллект: в 3 кн. / под ред. Э. В. Попова. М. : Радио и связь, 1990. Кн. 1: Системы общения и экспертные системы. 464 с.
- [6] Нелюбин Л. Л. Компьютерная лингвистика и машинный перевод / Л. Л. Нелюбин. М.: ВЦП, 1991. 151 с.
- [7] Попов Э. В. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта / Э. В. Попов, Г. Р. Фирдман. М. : Изд-во «Наука», 1976.-456 с.
- [8] Рассел С. Искусственный интеллект: современный подход: пер. с англ. / С. Рассел, П. Норвиг. 2-е изд. М. : Издательский дом «Вильяме», 2006.-1408 с.: ил.
- [9] Тетради новых терминов №6 (Англо-русские патентные термины). 2-е изд., испр. и доп. Москва, ВЦП, 1975. 434 с.
- [10] Федоров А. В. Основы общей теории перевода / А. В. Федоров. М. : Высш. шк., 1968.-286 с.
- [11] Флорин С. Муки переводческие / С. Флорин. М. : Высш. школа, 1983. 438 с.

Глава 10

ЗРИТЕЛЬНОЕ ВОСПРИЯТИЕ МИРА

10.1 Распознавание образов и обучение

10.1.1 Зрительное восприятие

Восприятие в искусственном интеллекте — процесс преобразования информации о внешнем мире, поступающей от анализаторной системы в форму для ее использования в интеллектуальной системе. Процесс восприятия обеспечивает связь системы со средой, создает и поддерживает внутреннюю модель мира, осуществляет планирование поведения и принятие решений реализует функции интеллектуального интерфейса.

Способы реализации процесса восприятия различны и зависят от характеристик поступающей информации и решаемой задачи. Итак, система восприятия— это переработка информации: предварительная обработка входных данных к стандартному виду, выделение семантически значимых информационных единиц слов, объектов, идентификация объектов и ситуаций, коррекция внутренней модели мира.

По типу регистрируемых анализаторной системой внешних сигналов можно говорить о восприятии зрительной информации (системы машинного зрения, технического зрения, экспертные системы анализа изображений), акустической информации (системы распознавания речи), тактильной информации (роботы) и т. д.

Анализаторная система в составе интеллектуальной системы — модуль, осуществляющий прием, накопление, предварительную фильтрацию всех внешних сигналов, как естественных (слуховые, визуальные, тактильные), так и искусственных (активные и пассивные излучения). Она является устройством ввода информации в подсистему восприятия интеллектуальной системы.

Восприятие зрительной информации — процесс извлечения из потока лучевой энергии, попадающего на рецепторы зрительной системы (искусственной или естественной), информации об окружающей среде. Результатом процесса является построение модели внешнего мира как в виде набора сведений об окружающей

среде (знания о направлениях, скоростях движения объектов), так и в образном виде (объемная модель наблюдаемой сцены). Такая система применяется для пространственной ориентации в окружающей среде, идентификации объектов обучения.

Зрительное восприятие как процесс моделирования среды. Физиологические и психологические данные говорят о том, что наше сознание способно не только создавать зрительные модели мира, субъективно неотличимые от реальности (например, во сне), но возможно постоянно делает это, проецируя «видимый образ» вовне, где он сливается со входным изображением наблюдаемого предмета.

Таким образом, вполне вероятно, что визуально воспринимаемый нами мир является фактически синтезируемой мозгом зрительной моделью реальности, управляемой поступающей из окружающей среды информацией и состоянием внутренних органов. Следовательно, процесс зрительного восприятия и понимания окружающей обстановки может быть смоделирован как неправильное сопоставление внешних видеоданных и внутренних моделей — прогнозов развития обстановки. Схема этого процесса представлена на рис. 10.1.

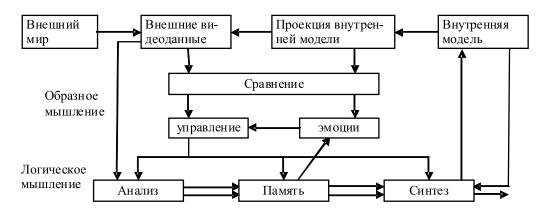
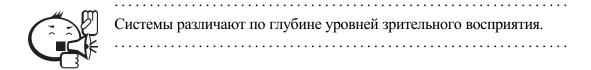


Рис. 10.1 – Процесс моделирования зрительного восприятия реальности.

Внешние данные поступают в модуль анализа, где проходят через каналы эталонного опознания, анализа элементов формы, пространственных и временных отношений [2].

Идентифицированные элементы реальности вызывают синтез соответствующих внутренних трехмерных представлений (моделей). Они должны учитывать собственные переменные «наблюдателя», а также экстраполировать движения объектов окружающей среды, иными словами — предсказывать состояние обстановки на некоторое время t вперед. Двумерная проекция предсказанного состояния обстановки сравнивается с текущим наблюдаемым образом. Результат сравнения влияет на общее состояние («эмоции») системы и режим дальнейшей работы.



Первый — получение изображения с рецепторов. В живых организмах от глаз в головном мозге, в технических системах реализуется путем подключения к ЭВМ

устройства цифрового ввода изображения, когда яркость каждой точки кодируется числами в ЭВМ.

Второй — построение образной модели. У животных и человека реализуется бессознательно и без участия внимания. В результате происходит разбиение зрительного поля на семантически значимые единицы (текста на буквы, сцены на объекты). Для технических систем разработано множество алгоритмов выделения и обнаружения объектов, сегментации изображений.

Третий — построение образно-семантической модели.

В системах искусственного интеллекта можно рассматривать два вида зрения:



Машинное зрение— направление исследований, ставящее своей целью моделирование зрительной информации восприятия, а в процессе функционирования накапливает образные и логические знания.

Машинное зрение можно рассматривать и как имитацию и тренинг интеллектуальной деятельности человека.

Техническое зрение имеет узкоспециализированное назначение и содержит минимально необходимый набор датчиков информации и средств обработки для выполнения заданных функций.

10.1.2 Основные сведения о распознавании образов

Существует множество задач искусственного интеллекта, решение которых связано с автоматизацией процессов восприятия и распознавания образов. Например, распознавание речи, анализ изображений и сцен, идентификация объектов, чтение рукописных текстов и т. д.

Распознавание образов непосредственно связанно с обучением, которое осуществляется на основе анализа и обобщения имеющихся данных.

Задачи распознавания (классификации) возникают в связи с обработкой и преобразованием на ЭВМ структур, представляющих в программах искусственного интеллекта знания о предметной области конкретной задаче.

Воспринимаемые объекты ПО обычно разбиваются на группы похожих, но не тождественных объектов. Множество объектов, обладающих общими свойствами, называют образом. Объекты относящиеся к одному образу (классу), отличаются второстепенными, несущественными с точки зрения решаемой задачи, свойствами.

Каждый образ (класс) представляется некоторым количеством структурированных объектов (экземпляров).

Общее свойство объектов, объединенных в образ (класс), называется признаками образа (класса) или описанием и необходимо для установления соответствия образов, т. е. доказательства их идентичности, аналогичности, сходства, осуществляемого сопоставлением.

В классических моделях образ описывается вектором признаков, в структурной модели образ это высказывание, порождаемое грамматикой, характеризующей

класс, которому данный образ принадлежит.

Образы формируются из простых стандартных элементов (абстрактные символы, множества, отношения или функции).

Отметим диалектическое единство понятий «образ» и «признак». Задание образов порождает признаки, а задание признаков порождает образы. Для выделения признаков сигналы, получаемые от различных рецепторов (датчиков), подвергаются различным преобразованиям (рис. 10.2) [10].



Рис. 10.2 – Обобщенная схема системы распознавания.

Во многих случаях непосредственно по сигналам, действующим на входе датчиков, невозможно выяснить классификацию объектов. Поэтому из этих сигналов выделяют необходимые признаки, которые формируют описание объекта в пространстве признаков. С помощью полученного описания выполняется распознавание объекта, т. е. отнесение его к тому или иному классу. Следовательно, решение задачи распознавания образов в значительной степени зависит от выбора множества признаков, обеспечивающих компактное представление объектов одного класса в пространстве признаков. Если указанные признаки выделены, то задачу распознавания образов можно рассматривать как задачу построения разделяющей поверхности, отделяющей одно компактное понятие от другого.

В тех случаях, когда имеющиеся априорные данные о распознаваемых объектах недостаточно полны, чтобы найти разделяющую поверхность, применяют обучаемые системы распознавания образов. Обучение осуществляется путем показа примеров с указанием принадлежности объектов к тому или иному классу (обучение с учителем) или без указания (обучение без учителя). В последнем случае система определяет множество признаков, на основе которых можно классифицировать объекты.

Д. В. Гаскаров распознавание связывает с двумя функциями [6]:

- отнесение объекта к неизвестному классификатору классу, т. е. процесс выделения новых классов кластеризация;
- идентификация объекта в качестве элементов известного классификатору класса, т. е. распознавание.

Целью создания автоматизированных вычислительных систем распознавания является автоматизация группы процессов восприятия и познания, связанных с поиском, выделением, идентификацией, классификацией и описанием образов на основе анализа реальных данных.

Поиск и выделение образов обычно являются начальным этапом анализа процесса обработки исходных данных.

Классификация — это анализ выборочных (преобразованных) данных; синтез модели, учитывающий отклонение образов; выбор из характеристик наиболее информативных, разработка алгоритма распознавания. Общая структура системы распознавания и ее разработка представлены на рисунке 10.3:

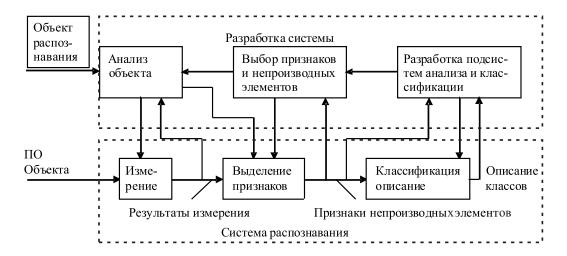


Рис. 10.3 – Общая структура системы распознавания и ее разработка.

Индуктивный вывод, или обобщение, основан на построении некоторого общего правила (закона) на основании конечного числа наблюдаемых фактов. В основе процесса индуктивного формирования понятий лежат умение человека выделять некоторые наиболее общие или характерные фрагменты среди описаний отдельных примеров понятий, избавляясь от мелких, незначительных характеристик, присущих конкретным примерам понятия. Создание упрощенной модели называется индуктивным обучением. В этом случае группируются сходные объекты в классы и конструируются правила, предсказывающие поведение объектов такого класса.

Изучение и компьютерное моделирование процесса обучения является одним из предметов исследования в области искусственного интеллекта, называемой машинным обучением. Здесь можно рассматривать добычу данных из БД (Data Mining) и обнаружение знаний, представленных в БД (Knowledge Discovery in Databases). Эти средства и системы могут применяться в машинном обучении распознавания образов и других задач.

В. Н. Вагин в работах по индуктивным методам использует семантически очень близкие термины классификации и распознавания в следующей трактовке[5]:

Под задачей распознания будем понимать в первую очередь задачу формирования решающего правила, на основании которого объекты, принадлежащие классу, могут быть отделены от остальных. По сути, когда мы говорим о получении решающего правила, речь идет о создании алгоритма, способного эффективно отличать объекты, принадлежащие классу, от прочих.

Решающее правило может быть задано по-разному. Прежде всего, любое решающее правило — это выбор условий, которым должны удовлетворять примеры формируемого понятия. Способ задания решающего правила может быть различным (логическое выражение, дерево решений, набор продукционных правил и т. д.).

Под задачей классификации будем понимать задачу отнесения вновь предъявленных объектов, отличных от объектов обучающей выборки, к тому или иному классу на основе построенного решающего правила. Речь идет о проверке полезности получаемого решающего правила на множестве контрольных примеров.

Рассматриваются два метода обучения. В обучении с учителем учитель опре-

деляет классы и предлагает примеры объектов каждого класса, сформировав тем самым описание класса. Описание класса вместе с самим классом дает нам классифицирующее правило «Если <описание> то <класс>, которое можно использовать для предсказания класса объектов, неизвестных системе ранее. Другой метод обучения — обучение без учителя — предполагает, что системе предъявляется совокупность объектов, не разделенных на классы. Количество классов также может быть неизвестным. Система должна сама определить классы объектов, основываясь на общих свойствах объектов из данного множества примеров.

Способы представления исходной информации в ИС. Из рис. 10.2 следует, что ЭВМ получает описание входной информации от внешних рецепторов в виде различных наборов данных, но в численной форме. Для отнесения ситуации к классу (понятию) необходимо иметь примеры ситуаций, используя обучающее множество.

Источником примеров для обучения может быть:

- учитель, задающий обучающие выборки;
- внешняя среда, когда обучающие выборки формируются случайным образом;
- интеллектуальная система, выполняющая обобщение.

Мы будем рассматривать только признаковые описания объекта, заданные набором признаков. Описание различных объектов отличается значениями признаков. Признаками могут быть объем, размер, вес, цена и т. д. Значение признаков объекта могут быть количественные, качественные и шкалированные.

Если заданы числовые признаки, то на множестве их значений можно ввести метрику, т. е. сравнивать их между собой.

Если заданы качественные признаки, то можно ввести ранговую шкалу (например: высокий, средний, маленький).

Задача распознавания — это отнесение объекта к одному из классов признаков. Для этого необходимо сравнение объектов и одним из способов сравнения является мера близости (сходства). Если известны объекты некоторого класса, то можно оценить меру близости объекта классу на основании значений меры близости данного объекта и объекта, принадлежащих классу. Мера близости должна быть максимальна.

Для числовых объектов в качестве меры близости обычно выбирается расстояние D(X,Y), где X и Y-два объекта, а x_k , y_k- соответствующие значения k-го признака объектов [5]:

$$D_{1}(X,Y) = \sqrt{\sum_{k=1}^{n} |(x_{k} - y_{k})^{2}|},$$

$$D_{2}(X,Y) = \sum_{k=1}^{n} |x_{k} - y_{k}|,$$

$$D_{3}(X,Y) = \max_{k} |x_{k} - y_{k}|.$$

Рассмотрим подробнее различные переходы к решению задачи об отнесении объекта к некоторому классу.

- Задача распознавания, т.е. система без обучения. Здесь известны классы и критерии отнесения объекта к классу (например, контроль качества детали).
- Задача распознавания, т. е. системы с обучением, когда не задан алгоритм разделения предъявляемых объектов на классы. Обучение и есть разработка этого алгоритма:
 - а) исходная информация (обучающее множество) не разделена на классы. Задача обучения сводится к разделению объектов выборки на группы «сходных» объектов (классы). Тогда процесс построения алгоритма, разделяющего затем объекты по степени их сходства, называется обучением «без учителя»;
 - б) исходная информация представлена в виде множества объектов с указанием, к какому классу он относится, но критерий нам не известен. Задача обучения заключается в построении алгоритма, который позволяет предъявленный объект отнести к заданному классу. Тогда процесс построения такого алгоритма назовем обучением «с учителем».

10.1.3 Геометрический метод распознавания

Задача обучения без учителя. Формулировка задачи. Пусть обучающая выборка содержит M объектов $X = \{x_1, x_2, \dots, x_M\}$. Каждый из этих объектов представляет собой n-мерный вектор X_i значений признаков:

$$X_i:\langle x_{i1},x_{i2},\ldots,x_{in}\rangle,$$

где x_{ij} — значение j-го признака для i-го объекта,

n — количество признаков, характеризующих объект.

Требуется в соответствии с заданным критерием разделить набор X на классы, количество которых заранее неизвестно. Под критерием понимается мера близости численных признаков всех объектов одного класса.

Представим алгоритм, основанный на понятии порогового расстояния.

Критерием отнесения объекта к классу в этом алгоритме является пороговое расстояние T. Если объект находится в пределах порогового расстояния от точки-прототипа (типичной точки класса по какому-либо критерию) некоторого класса, то такой объект будет отнесен к данному классу, если нет, то он становится прототипом нового класса.

Алгоритм.

- 1) Выбрать точку-прототип первого класса (например, объект X_1 из обучающей выборки). Количество классов K положить равным 1. Обозначить точку-прототип Z_1 .
- 2) Определить наиболее удаленный от Z_1 объект X_f по условию $D\left(Z,X_f\right)=\max_i D\left(Z_1,X_i\right)$, где $D\left(Z_1,X_i\right)$ —расстояние между z_1 и x_i , вычисленное одним из возможных способов. Объявить X_f прототипом второго класса. Обозначить X_f как Z_2 . Число классов k=k+1.

- 3) Определить пороговое расстояние $T = D(Z_1, X_f)/2$. Построить $X' = X \setminus \{X_1, X_f\}$.
- 4) Выбрать $X_f \in X'$.
- 5) Вычислить расстояние от X_j до всех точек прототипов: $D(Z_k, X_j)$ для k = 1, 2, ..., K.
- 6) Определить ближайшую к рассматриваемому объекту точку-прототипа Z_p по условию $D\left(Z_p,X_j\right)=\min_i D\left(Z_k,X_j\right)$.
- 7) Если $D(Z_p, X_j) < T$, отнести объект X_j к классу $p(Z_p)$ является прототипом этого класса) и удалить его из X'. Иначе объявить X_j прототипом нового класса. Обозначить X_j как Z_{k+1} . Число классов k увеличить на 1: k = k+1. Удалить X_j из X?.
- 8) Если X' = 0, т. е. обучающее множество исчерпано, то КОНЕЦ. В противном случае перейти к шагу 4).



Пусть каждый объект из множества объектов, представленных в табл. 10.1, задан двумя признаками, а модель — точке на плоскости на рис. 10.4

Таблица 10.1 – Координаты точек обучающей выборки и их расстояние.

	X1	X2	X3	X4	X5	X6	X7	X8
Координаты объек-	(2; 2)	(3; 3)	(2; 4)	(7; 1)	(8; 5)	(9; 1)	(9; 2)	(9; 6)
ТОВ								
Точки-прототипы	Z1			Z3				Z2
классов								
Расстояние от Z1		1.4	2	5.1	6.7	7.05	7.0	8.05
до объектов								
Расстояние от Z2		6.7	7.3	5.4	1.4	5	4	
до объектов								
Расстояние от Z3					4.1	2	2.2	
до объекта								

Окончательное распределение представлено в табл. 10.2.

Таблица 10.2 – Распределение элементов по классам.

	Прототип	Элементы класса		
Класс 1	X1 (2, 2)	X2 (3; 3)	X3 (2; 4)	
Класс 2	X8 (9; 6)	X5 (8; 5)		
Класс 3	X4 (7; 1)	X6 (9; 1)	X7 (9; 2)	

Алгоритмы MAXMIN, «К средних», алгоритмы с использованием решающих функций по критерию минимального расстояния, разделяющие решающие

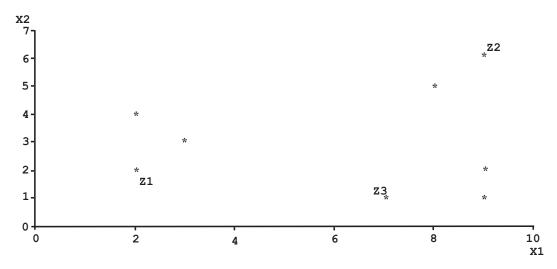


Рис. 10.4 – Точки обучающей выборки и их расстояний на плоскости.

функции, линейные решающие функции, метод потенциалов, распознавание на основе приближенных признаков можно рассмотреть в замечательной книге В. Н. Вагина.

Общая идея, лежащая в основе классификации по минимуму расстояния, предполагает, что все точки, представляющие объекты одного класса в пространстве признаков, расположены близко друг к другу, а точки различных классов удалены друг от друга на некоторое расстояние. В этом случае улучшить результаты классификации можно, если минимизировать расстояние между точками одного класса и максимизировать расстояние между точками разных классов. С этой целью могут выполняться линейные и нелинейные преобразования, деформирующие пространство признаков. Из-за сложности вычислений геометрические методы распознавания используют в основном для иллюстрации других методов.

10.1.4 Байесовский метод распознавания

Байесовский метод распознавания объектов [4] предусматривает построение решающих правил на основе статистических свойств классов. Пусть \overrightarrow{x} — вектор признаков, соответствующий k_i классу. Априорная вероятность реализации класса k_i равна $P(k_i)$. Задача состоит в том, чтобы отнести входной вектор \overrightarrow{x} к одному из известных классов с минимальной ошибкой. Считается, что система распознавания допускает ошибку, если относит к классу k_j объект, на самом деле принадлежащий классу k_i .

Если в результате измерений определить $p\left(\overrightarrow{x}|k_i\right)$ — плотность распределения вектора \overrightarrow{x} при условии, что он принадлежит классу k_i , то в соответствии с теоремой Байеса:

$$p\left(\overrightarrow{x}\left|k_{i}\right.\right) = \frac{p\left(\overrightarrow{x}\left|k_{j}\right.\right) * P\left(k_{i}\right.\right)}{\sum_{j=1}^{M} p\left(\overrightarrow{x}''k_{j}\right) * P\left(k_{j}\right)},$$

где M — число классов. Это выражение определяет вероятность того, что \vec{x} при-

надлежит классу k_i , а отнесение вектора \vec{x} к классу k_i можно считать оправданным, если выполняется условие:

$$p(k_i|\overrightarrow{x}) \geqslant p(k_i|\overrightarrow{x}), \forall j \neq i.$$



Данное правило называется правилом максимума апостериорной вероятности.

Из двух вышеприведенных формул следует байесовское решающее правило:

$$p(\overrightarrow{x}|k_i)P(k_i) \ge p(\overrightarrow{x}|k_j)P(k_j), \forall j = i.$$

В частном случае, когда априорные вероятности реализации классов равны $P(k_i) = P(k_j)$ решающее правило представляется в виде:

$$p(\overrightarrow{x}|k_i) \geqslant p(\overrightarrow{x}|k_j), \ \forall j \neq i.$$

Или

$$\frac{p\left(\overrightarrow{x}|k_{i}\right)}{p\left(\overrightarrow{x}|k_{i}\right)} \geqslant 1, \ \forall j \neq i.$$

Функция $p(\vec{x}|k_i)$ называется функцией правдоподобия, а последнее правило — правилом максимального правдоподобия.

Можно ввести функцию потерь $C(k_i,k_j)$, которая характеризует потери (риск), возникающие при отнесении объекта, принадлежащего класса k_i , к классу k_j . Математическое ожидание потерь, связанных с отнесением \overrightarrow{x} к классу k_j , определяется из выражения:

$$R_{j}(\overrightarrow{x}) = \sum_{i=1}^{M} C(k_{i}, k_{j}) p(k_{i}|\overrightarrow{x}).$$

Величину $R_j(\vec{x})$ называют *условным средним риском* или условными средними потерями. Общие средние потери при выбора класса k_j равны

$$R_{j} = \int_{Y} R_{j}(\overrightarrow{x}) p(\overrightarrow{x}) d\overrightarrow{x},$$

где X — область определения \overrightarrow{x} .

Классификация на основе байесовского решающего правила обеспечивает минимизацию общих средних потерь, называемых среднем риском (без доказательства).

Применение байесовского подхода к решению задачи классификации требует знания априорных вероятностей появления классов $P(k_i)$, плотностей распределения вероятностей для каждого из классов $p(\overrightarrow{x}|k_i)$, а так же соответствующих значений функций потерь.

10.1.5 Синтаксический метод распознавания

Предыдущие методы распознавания основывались на разделении объектов в пространстве признаков. Синтаксический подход базируется на использовании структурных отношений, характеризующих распознаваемый образ. Здесь в распознаваемом объекте выделяют простейшие структурные элементы и выполняют описание сложного объекта в виде иерархической структуры, состоящей из простых элементов. Типичным примером задач, где применяется такой подход, является распознавание изображений и, в частности, анализ сцен [1].

Синтаксический метод распознавания (структурные отношения) заключается в том, что выделяют не признаки, а простейшие структурные элементы и получают описание сложного объекта в виде иерархической структуры.

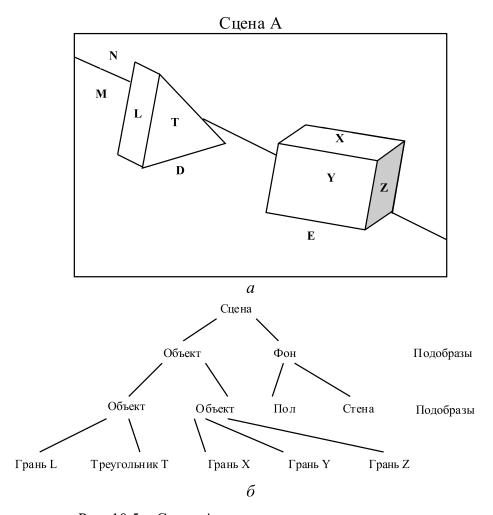


Рис. 10.5 – Сцена А и ее иерархическое описание.

В качестве примера рассмотрим сцену, изображенную на рис. 10.5, a. Иерархическое описание данной сцены представлено в виде дерева на рис. 10.5, δ . Здесь сцена А состоит из подобразов, часть которых представляется элементарными подобразами — гранями, треугольником. Сцена определяется как некоторая структура, состоящая из подобразов, аналогично тому, как предложения естественного языка состоят из слов при помощи грамматических правил, а слово образуется из

букв. Благодаря этой аналогии рассматриваемый подход называется синтаксическим (лингвистическим, структурным) подходом к распознаванию образов.

Процесс распознавания образов заключается в следующем:

- выделение простейших элементов примитивов;
- на основе выделенных примитивов строится структурное описание объекта распознавания;
- выполнение синтаксического анализа, в ходе которого устанавливается, является ли это описание объекта синтаксически правильным относительно заданной грамматики;
- зачисление объекта в тот класс, в котором его описание оказывается синтаксически правильным.

Структурная схема системы синтаксического распознавания образов изображена на рис. 10.6.

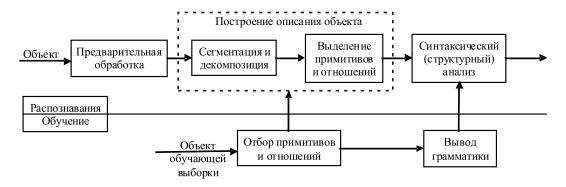


Рис. 10.6 – Система синтаксического распознавания образов.

На этапе предварительной обработки данные об объекте распознавания с целью улучшения качества подвергаются фильтрации, аппроксимации, восстановлению и т. д. Процесс построения описания объекта предполагает выполнение двух процедур: сегментации на подобъекты и выделения примитивов в соответствии с заранее определенными синтаксическими операциями. В результате каждый объект представляется с помощью некоторого набора примитивов и фиксированных синтаксических операций. Если используется операция конкатенации, то объект представляется строкой, состоящей из последовательности примитивов.

На этапе синтаксического анализа могут выполняться либо простейшее сопоставление с эталоном (образцом), либо полный грамматический разбор. В случае с сопоставлением с эталоном строка примитивов, представляющая исходный объект, сравнивается со строками, представляющими объект-прототип. В соответствии с выбранным критерием подобия строк объект зачисляется в тот класс, к которому относится объект-прототип. Информация об иерархической структуре объекта в этом случае игнорируется. Полный грамматический разбор строки, описывающий распознаваемый объект, позволяет выявить особенности структуры объекта и учесть их при принятии решения.

Вывод, грамматически необходимый для выполнения синтаксического анализа, осуществляется по заданной обучающей выборке объектов. При этом отбор примитивов и структурных отношений во многих случаях выполняется самим

проектировщиком системы распознавания. Вывод грамматики, по сути, представляет обучение синтаксической системы распознавания.

Формальные грамматики. Будем называть языком конечное или бесконечное множество предложений, каждое из которых имеет конечную длину и построено с помощью операции соединения из конечного множества элементов. Это определение включает как естественные, так и искусственные языки логики и программирования.

Чтобы точно определить язык, необходимо установить общие принципы, которые отделяют последовательности атомарных элементов, являющихся предложениями, от последовательностей, таковыми не являющихся. Это различие нельзя выразить просто списком, так как для всех систем, представляющих интерес, не накладывается ограничений по количеству предложений и их длине. Традиционные грамматики естественных языков не содержат всей информации, необходимой для выполнения указанной задачи. Стремление к более строгой лингвистической теории привело к разработке формальных грамматик, основоположником которых по праву считается Н. Хомский. Он указал на возможность использования для описания естественных языков некоторых исчислений, рассматривавшихся в теории алгоритмов.

Требования к лингвистической теории.

- 1) Порождать все те, и только те предложения, которые составляют описываемый данной грамматикой язык.
- 2) Определять метод, с помощью которого можно было бы дать единственное и единообразное структурное описание каждого предложения, порожденного грамматикой.

Основные понятия теории формальных грамматик [1]. Пусть V — непустое конечное множество, которое мы будем называть словарем (алфавитом), а его элементы символами (буквами). Грамматика G — это четверка

$$G = (V_N, V_T, P, S)$$
,

где V_N и V_T — конечные множества начальных и терминальных символов соответственно, $V_N \cap V_T \neq 0$ и $V_N \cup V_T = V$; P — конечное множество продукционных правил $\alpha \to \beta$, где α и β представляют строки символов из V, причем α содержит, по крайней мере, один символ из V_N ; $S \in V_N$ — начальный символ. Множество V_T называется терминальным (основным) словарем, а множество V_N — нетерминальным (вспомогательным) словарем. Терминальный словарь — это набор исходных элементов, из которых строятся цепочки символов (строки), порождаемые грамматикой. Нетерминальный словарь — это набор символов, которым обозначаются классы цепочек из исходных элементов. Начальный символ S представляет собой элемент множества V_N и обозначает класс всех тех объектов, для описания которых и предназначена грамматика.

Для произвольного словаря V обозначение V^* будем использовать для идентификации всех строк, составленных из V, включая пустую строку λ . Обозначение V^+ будет соответствовать множеству строк из V без пустой строки, т. е. $V^+ = V^* - \{\lambda\}$. Например, пусть $V = \{a,b\}$, тогда $V^* = \{\lambda,a,b,aa,ab,ba,\dots\}$ и $V^+ = \{a,b,aa,ab,\dots\}$.

Язык, порождаемый грамматикой G и обозначаемый L(G), представляет множество строк (цепочек):

$$L(G) = \{x | x \in V_T^* \text{ if } S^* \underset{G}{\Rightarrow} x\}.$$

Каждая из строк языка состоит только из терминальных символов. При этом любая строка может быть выведена из S путем применения правил из P, что обозначено $S^* \Rightarrow x$.

Грамматики различают по типу правил подстановки, допустимых в каждой из них.

Контекстно-зависимая грамматика характеризуется правилами подстановки вида $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$, где α_1 и α_2 —элементы словаря $V^*, \beta \in V^+, A \in V_N$. Такая грамматика позволяет выполнить замещение нетерминального символа строкой β , если A появляется в контексте α_1 и α_2 .

Контекстно-свободная грамматика характеризуется правилами подстановки вида $A \to \beta$. Здесь символ $A \in V_N$ может замещаться строкой β независимо от контекста, в котором появляется A.

Грамматика называется **регулярной**, если используются правила подстановки вида $A \to aB$ или $A \to a$, где $A, B, \in V_N, a \in V_T$. При этом альтернативными правилами подстановки являются $A \to Ba$ и $A \to a$. Однако выбор одной группы правил исключает применение другой группы.

В качестве примера рассмотрим контекстно-зависимую грамматику $G_0 = (V_N, V_T, P, S)$, где $V_N = \{S, A, B\}$, $V_T = \{a, b, c\}$ и множество P заданно правилами продукции:

- 1) $S \rightarrow aSBA$
- 2) $S \rightarrow abA$
- 3) $AB \rightarrow BA$
- 4) $bB \rightarrow BB$
- 5) $bA \rightarrow bc$
- 6) $cA \rightarrow cc$
- 7) $aB \rightarrow ab$

Язык L(G), порождаемый грамматикой G_0 , является множеством всех строк, состоящих из одинаковых чисел символов a,b,c.

$$L(G) = \{a^n b^n c^n | n \ge 1\},\,$$

Например,

$$S \stackrel{(2)}{\Rightarrow} abA \stackrel{(5)}{\Rightarrow} abc$$
 или

$$S \stackrel{(1)}{\Rightarrow} aSBA \stackrel{(2)}{\Rightarrow} aabABA \stackrel{(3)}{\Rightarrow} aabBAA \stackrel{(4)}{\Rightarrow} aabbAA \stackrel{(5)}{\Rightarrow} aabbcA \stackrel{(6)}{\Rightarrow} aabbcc.$$

Здесь числа в скобках указывают номера использованных продукций. Если символы a,b,c будут соответствовать графическим примитивам:

 $a: \to; b: \nwarrow; c: \swarrow;$ то грамматика $L(G_0)$ будет порождать строки, соответствующие равносторонним треугольникам (рис. 10.7).



Рис. 10.7 – Множество фигур, порожденных грамматикой $L(G_0)$.

Значение формальных грамматик для распознавания состоит в том, что они позволяют построить алгоритм для ответа на вопрос о синтаксической правильности или неправильности представления объекта с помощью примитивов. Этот алгоритм представляет процедуру синтаксического анализа (грамматического разбора). Существует два основных вида синтаксического анализа: разбор сверху вниз и разбор снизу вверх. В процедуре разбора сверху вниз правила-продукции применяются к начальному символу S, чтобы получить заданную строку, представляющую распознаваемый объект. Данная процедура реализует процесс подстановок, управляемый целью. При разборе снизу вверх продукционные правила применяются в обратном направлении с целью приведения строки, представляющей распознаваемый объект, к начальному символу. Прямые реализации указанных схем грамматического разбора неэффективны, так как требуют выполнения полного перебора.

Важной подзадачей, возникающей в ходе применения синтаксического метода распознавания, является построение (конструирования, вывод) грамматики по обучающей выборке. Конструирование грамматики является лингвистическим эквивалентом алгоритмов обучения. Универсальных автоматических методов построения требуемой грамматики по примерам не существует из-за ограничений на предметные области.

10.2 Компьютерное зрение

Компьютерное зрение (Computer vision) (машинное зрение, техническое зрение, зрение роботов) оформилось как самостоятельная дисциплина к концу 60-х годов. Это направление возникло в рамках искусственного интеллекта в тот его период, когда еще были горячи споры о возможности создания мыслящей машины. Оно выделилось из работ по распознаванию образов, когда была осознана его математическая специфика, фотоизображение представляло собой двухмерную проекцию трехмерного мира, т. е. вырожденное преобразование.

Фотоизображение начали использовать для измерений задолго до изобретения компьютера (фотограмметрия). Этот подход использовался для аэрофотосъемки, позже — для космической навигации и других задач.

С развитием техники, особенно с появлением в 90-е годы персональных компьютеров и цифровых фото- и видеокамер, наступила новая, современная эра в области машинного зрения. Проблемы получения информации, сопряжения оптиче-

ского канала ввода с ЭВМ и визуализация были решены, а область содержательных исследований зрительных проблем, а также применения видеоввода для измерений и автоматического наблюдения быстро расширяется.

10.2.1 Типичные задачи компьютерного зрения

Каждая из областей применения компьютерного зрения связана с рядом задач. Некоторые примеры типичных задач компьютерного зрения представлены ниже [12].

Распознавание. Классическая задача компьютерного зрения — определение некоторого характерного объекта с его особенностями и активностью в видеоданных при заданных условиях внешней среды. Эта задача может быть достоверно и легко решена человеком, но до сих пор не решена удовлетворительно в компьютерном зрении в общем случае: случайные объекты в случайных ситуациях.

Существующие методы для решения таких проблем могут быть достоверно решены только для отдельных объектов, таких как простые геометрические объекты (например, многогранники), человеческие лица, печатные или рукописные символы, автомобили и только в определенных условиях, обычно это определенное освещение, фон и положение объекта относительно камеры. В литературе описано различное множество проблем распознавания:

- Распознавание: один или несколько предварительно заданных или изученных объектов или классов объектов могут быть распознаны, обычно вместе с их двухмерным положением на изображении или трехмерным положением на сцене.
- Идентификация: распознается индивидуальный экземпляр объекта. Примеры: идентификация определенного человеческого лица или отпечатка пальцев или автомобиля.
- Обнаружение: видеоданные проверяются на наличие определенного условия. Например, обнаружение возможных неправильных клеток или тканей в медицинских изображениях. Обнаружение, основанное на относительно простых и быстрых вычислениях, иногда используется для нахождения небольших участков в анализируемом изображении, которые затем анализируются с помощью приемов, более требовательных к ресурсам, для получения правильной интерпретации.

Существует несколько специализированных задач, основанных на распознавании текстов, например:

- Поиск изображений по содержанию: нахождение всех изображений в большом наборе изображений, которые имеют определенное содержание. Содержание может быть определено различными путями, например в терминах схожести с конкретным изображением (найдите мне все изображения, похожие на данное изображение), или в терминах высокоуровневых критериев поиска, вводимых как тестовые данные (найдите мне все изображения, на которых изображено много домов, которые сделаны зимой и на которых нет машин).
- Оценка положения: определение положения или ориентации определенного объекта относительно камеры. Примером применения этой техники

может быть содействие руке робота в извлечении объектов с ленты конвейера на линии сборки.

• Оптическое распознавание знаков: распознавание символов на изображениях печатного или рукописного текста, обычно для перевода в текстовый формат, наиболее удобный для редактирования или индексации (например, ASCII).

Движение. Несколько задач, связанных с оценкой движения, в которых последовательность изображений (видеоданные) обрабатываются для нахождения оценки скорости каждой точки изображения или 3D-сцены.

Примерами таких задач являются:

- определение трехмерного движения камеры;
- слежение, т. е. следование за перемещениями объекта (например, машин или людей).

Восстановление сцены. Даны два или больше изображений сцены, или видеоданные. Восстановление сцены имеет задачей воссоздать трехмерную модель сцены. В простейшем случае моделью может быть набор точек трехмерного пространства. Более сложные методы воспроизводят полную трехмерную модель.

Восстановление изображений. Задача восстановления изображений — это удаление шума (шум датчика, размытость движущегося объекта и т. д.). Наиболее простым подходом к решению этой задачи являются различные типы фильтров, таких, как фильтры нижних или средних частот. Более сложные методы используют представления того, как должны выглядеть те или иные участки изображения, и, на основе этого, их изменение.

Более высокий уровень удаления шумов достигается в ходе первоначального анализа видеоданных на наличие различных структур, таких, как линии или границы, а затем управления процессом фильтрации на основе этих данных.

10.2.2 Системы компьютерного зрения

Реализация систем компьютерного зрения сильно зависит от области их применения. Некоторые системы являются автономными и решают специфические проблемы детектирования и измерения, тогда как другие системы составляют подсистемы более крупных систем, которые, например, могут содержать подсистемы контроля за механическими манипуляторами, планирования, информационные базы данных, интерфейсы человек-машина и т. д. Реализация систем компьютерного зрения также зависит от того, является ли ее функциональность заранее определенной или некоторые ее части могут быть изучены и модифицированы в процессе работы. Однако существуют функции, типичные для многих систем компьютерного зрения.

• Видеосъемка: цифровые изображения производятся посредством одного или нескольких датчиков изображения, которые, за исключением различных типов светочувствительных камер, включают датчики расстояния, радары, ультразвуковые камеры и т. д. в зависимости от типа датчика, получающиеся данные могут быть обычными 2D-изображением, 3D-изображением или последовательностью изображений. Значения пикселей обычно со-

ответствуют интенсивности света в одной или нескольких спектральных полосах (цветные или изображения в оттенках серого), но могут быть связаны с различными физическими измерениями, такими, как глубина, поглощение или отражение звуковых или электромагнитных волн, или ядерным магнитным резонансом.

- Предварительная обработка: перед тем как методы компьютерного зрения могут быть применены к видеоданным, с тем чтобы извлечь определенную долю информации, необходимо обработать видеоданные, с тем чтобы они удовлетворяли некоторым условиям, в зависимости от используемого метода. Примерами являются:
 - повторная выборка, с тем чтобы убедиться, что координатная система изображения верна;
 - удаление шума, с тем чтобы удалить искажения, вносимые датчиком;
 - улучшение контрастности, для того чтобы нужная информация могла быть обнаружена;
 - масштабирование для лучшего различения структур на изображении.
- Выделение деталей: детали изображения различного уровня сложности выделяются из видеоданных. Типичными примерами таких деталей являются:
 - линии, границы и кромки;
 - локализированные точки интереса, такие, как углы, капли или точки: более сложные детали могут относиться к структуре, форме или движению.
- Детектирование/Сегментация: на определенном этапе обработки принимается решение о том, какие точки или участки изображения являются важными для дальнейшей обработки. Примерами являются:
 - выделение определенного набора интересующих точек;
 - сегментация одного или нескольких участков изображения, которые содержат характерный объект.
- Высокоуровневая обработка: на этом шаге входные данные обычно представляют небольшой набор данных, например набор точек или участок изображения, в котором предположительно находится определенный объект. Примерами являются:
 - проверка того, что данные удовлетворяют условиям, зависящим от метода применения;
 - оценка характерных параметров, таких, как положение или размер объекта:
 - классификация обнаруженного объекта по различным категориям.

Конечная цель системы (подсистемы) компьютерного зрения — формирование символьного описания изображения, т. е. его понимание; а исследования — направлены на изучение методов понимания изображения. На основе одного или

нескольких двумерных изображений система компьютерного зрения формирует символьное описание реальности, представляемой изображением.



Если представляемая изображение — реальность — часть трехмерного мира, то следует говорить о трехмерном компьютерном зрении или анализе сцен.

.....

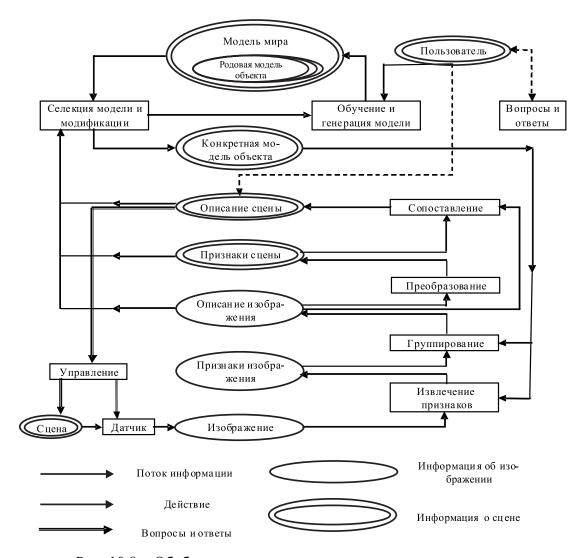


Рис. 10.8 – Обобщенная схема системы компьютерного зрения.

Изображения, анализируемые системами компьютерного зрения, могут очень отличаться по своей природе: двумерные черно-белые и цветные; множественные изображения в случае стереозрения; диапазонные изображения (с помощью датчиков глубины); трехмерные изображения и т. д.

Идея разработки обобщенной системы компьютерного зрения нашла воплощение в конце 80-х годов: VISION, ACRONYM, SIGMA.

10.2.3 Этапы понимания изображения

- Реализация в интеллектуальных системах трех основных функций зрения и ориентации, узнавания, понимания связаны с установлением соответствия между элементами структуры, представляющей видеоданные в памяти, и элементами (фактами) базы знаний. Они представляются следуюшими этапами:
 - Выбор адекватного представления видеоданных.
 - Выбор формы представления знаний. Здесь можно использовать три типа знаний: базу знаний, создаваемую разработчиком и описывающую модель мира, в котором работает система; накапливаемые знания в виде классифицирующих образы признаков и новых понятий, описывающих классы образов; накапливаемые знания в виде конкретных образов и их элементов (частей). В системах восприятия представление видеоданных — это структура данных (логическая или физическая), с которой взаимодействуют аппаратные и программные компоненты системы обработки изображений и от которой зависит сложность и эффективность алгоритмов дальнейших преобразований видеоданных и построения образной иерархии. Удачно выбранная структура данных позволяет сократить объемы передаваемой и хранимой в системе информации, избежать трудоемких операций преобразования видеоданных из одной формы в другую. Наиболее часто в системах восприятия ИС используются следующие представления видеоданных:
 - матричное и прямое, где каждый элемент описывает яркость или цвет элемента изображения;
 - векторное, где каждый элемент описывает отрезок прямой линии определенной длины и ориентации;
 - синтаксическое, элементами которого являются алфавит символов (простые, часто встречающиеся на изображении конфигурации типа отрезка, дуги, узла и т. п.) и объекты, конструируемые по определенным правилам из этих символов;
 - на основе ортогональных преобразований, где каждый элемент описывает яркость и цвет области изображения определенной формы и различных размеров (от элемента яркости до изображения в целом).

На основе перечисленных форм и их комбинаций представляются и затем анализируются как двумерные, так и трехмерные изображения.

Развивается так же морфологическое описание, основанное на выводах интегральной геометрии и математической морфологии, преобразование Хоха, дающее удобные средства работы с линиями. В рамках этих представлений элементарные объекты имеют, помимо положения, и другие характеристики (например, размер, форму).

2) Следующий этап — переход от внутреннего представления видеоданных к образной иерархии, которую можно рассматривать как взаимосвязанный

набор признаков, деталей и частей образов, имеющихся на изображении. Обычно начинают с так называемой предобработки изображения — устранения шумов и искажений в сенсорных устройствах или в потоке видеоданных, поступивших в систему; выравнивания яркости и усиления (если необходимо) контрастности; других операций, не затрагивающих семантик изображения, но субъективно (для человека) или объективно (для выполнения последующих преобразований) улучшающих его качество.

Предварительная обработка изображений проводится для улучшения качества видеоданных и устранения искажений, возникающих в процессе получения и представления видеоданных в ЭВМ (например, из-за недостаточной прозрачности атмосферы при аэросъемке, движения снимаемого объекта, помех в каналах передачи данных, низкой чувствительности элементов сенсорной системы и т. п.). Задачу устранения искажений называют также реставрацией изображения. И улучшение качества, и реставрация изображения тесно связаны с задачей выделения информации из видеоданных, являясь первыми этапами анализа, на которых подготавливаются входные данные для алгоритмов выделения признаков.

Перечислим основные методы улучшения качества изображений и модели компенсации различных искажений:

- Изменение общей или локальной (на отдельных участках) яркости и (или) контрастности изображения на основе преобразования его гистограммы яркости.
- Подчеркивание границ (перепадов яркости) на изображении для обнаружения контуров или улучшения субъективного (с точки зрения человека) качества изображения.
- Подавление на изображении шумов и помех на основе различных видов фильтрации.
- Коррекция геометрических искажений на основе преобразования координат его элементов с целью изменения масштаба изображения, его поворота, сдвига, аффинных или проективных преобразований.
- Реставрация нерезких изображений устранение ошибок фокусировки оптической части сенсорной системы, рассеивающего влияния атмосферы.
- Реставрация смазанных (вследствие движения камеры или объекта) изображений.
- Компенсация искажений, вносимых зависимостью фотопленки, ограниченной разрешающей способностью устройств ввода изображений.

Преобразование значений яркостей элементов изображения часто применяется при малой яркости, недостаточной контрастности входного изображения, плохой различимости деталей «в тенях» или «в ответах». При этом гистограмма яркости имеет ярко выраженный пикообразный характер или охватывает лишь небольшую часть возможного диапазона яркостей. Для улучшения качества таких изображений применяются два основных метода: задание функции изменения шкалы яркости и «выравнивание» гистограммы (всего изображения или, возможно, в отдельных зонах с малой контрастностью). Выравнивание гистограммы означает приведение ее к примерно равномерному виду (возможно, с потерей некоторого числа уров-

ней квантования яркости), что обеспечивает улучшение субъективного восприятия изображения, повышает различимость деталей.

3) Базовые операции при анализе изображений [4]. Объект обычно выделяется по группировке пикселей и основывается на выделенных областях и границах (альтернативные варианты).

Выделение граничных элементов. Границы на изображении обнаруживаются путем вычисления градиента интенсивности, который характеризуется:

- направлением (вдоль или перпендикулярно границе);
- позицией;
- силой.

Контуры объекта обычно видны как некоторые границы на изображении. Поэтому естественным является стремление выделить объект по его границам. Но изменение отражающей поверхности или цвета поверхности объекта, тени или искажение пространственной ориентации увеличивают неопределенность. Поэтому границы распознают постепенно, от отдельных точек к сегментам границы.

Элементом границы является пиксель (англ. edgel — элемент границы). Элементы границы первоначально обнаруживаются при помощи градиентных (граничных) операторов (Собела, Кирша, Канни и др.). Они отличаются степенью подавления шумов и угловой разрешающей способностью.

Восстановление границ. Методы восстановления границ используют информацию о геометрическом соседстве точек изображения для присвоения им согласованных меток на основе ненадежных данных, формируемых датчиками границ. В конце постобработки формируется некоторая улучшенная граница изображения, элементы которой первоначально определяются с помощью соответствующего граничного оператора.

Граничные сегменты. Существуют три принципа группирования: коллинеарность, параллелизм и геометрическая близость. Используя их, можно отдельные элементы изображения объединить в нечто целое. Для этого существуют следующие метолы:

- Преобразования Хафа, которые позволяют обнаруживать на изображении кривые заданной формы (прямоугольник, круг, эллипс и т.п.). При этом рассматривают два пространства: пространство изображений и пространство параметров (рис. 10.9). Прямые линии пространства изображения представляются точками в пространстве параметров, а точки пространства изображений представляются прямыми в пространстве параметров:
- Метод поиска на графах при обнаружении контурных сегментов.
- Метод глобальной оптимизации выделяемого контура метод активных контуров («снейков»).

Выделение областей изображения. Подходы, основанные на выделении областей изображения, являются дуальными по отношению к подходами, основанным на выделении контуров: контуры ограничивают области, а соприкосновение различных областей между собой создают контуры. Отдельные пиксели могут играть роль исходных точек в процессе генерации областей. Пиксель добавляется к области, когда его специфические свойства не слишком отличаются от свойств

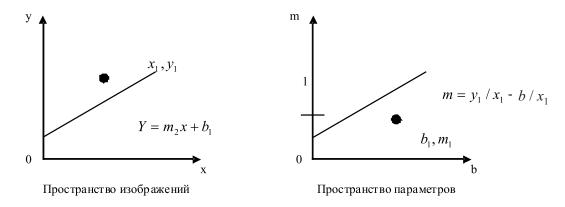


Рис. 10.9 – Преобразование Хафа для прямых линий.

пикселей, которые уже образуют область, и когда он соседствует с областью. Области, в которых интенсивность изменяется плавно, нужно рассматривать как одну. Данный процесс называют «слиянием», и он выполняется при некоторых ограничениях. Можно использовать и противоположный процесс — «разделения».

4) **Интерпретация контурных рисунков.** Рассмотрим метод интерпретации трехмерной сцены по двумерному контурному рисунку, т. е. классический метод Хаффмана и Клоуза, а также Вальца.

Для этого необходимо обладать знаниями об окружающей среде и об объектах и учитывать ограничения:

- контурные рисунки должны быть корректны (нет разрывов и свободных концов линий);
- объекты на рисунках должны быть видимы с общих позиций;
- объекты должны быть многогранниками, в каждом углу объекта пересекаются три плоскости.

Задача решается методом распространения ограничений. Необходимо различать элементы, принадлежащие двумерному изображению, и элементы соответствующей трехмерной сцены. Элементами 2D-изображения являются: линия, соединение, область. Элементами 3D-изображения являются: край (ребро), вершина, поверхность.

- 1) Классификация. Выделяют следующие виды двумерных элементов:
 - линия отрезок прямой \;
 - L-соединение пересечение двух линий <a>;
 - стрелка пересечение трех линий, наибольший угол между двумя линиями превышает 180° ↓ ;
 - вилка пересечение трех линий, угол между любыми двумя линиями меньше 180° Y;
 - Т-соединение пересечение трех линий, один угол равен 180° Т.
- 2) Осуществляется возможная интерпретация линий и соединений в трехмерном пространстве. Для линий вводятся метки:

- выпуклые ребра;
- вогнутые ребра;
- закрывающие ребра.

Множество интерпретаций соединений можно получить исходя из четырех типов пространственного размещения блоков, рассматриваемых под разным углом зрения.

10.3 Машинное зрение

Машинное зрение тесно соприкасается с тремя областями: обработкой изображений, классификацией образов и анализом сцен [7, 11, 12].

Обработка изображений в основном касается получения новых изображений по уже имеющимся. Большинство применяемых здесь методов заимствовано из теории линейных систем. На новом изображении можно подавить шум, устранить размывание или подчеркнуть границы. Окончательный результат представляет собой изображение, которое, как правило, предназначено для интерпретации человеком.

Основной задачей классификации образов является отнесение «образа» к тому или иному классу, причем образ обычно представляется в виде набора чисел, соответствующих измеренным характеристикам объекта (высоте, ширине). Хотя на вход классификатора подается не изображение, сами методы классификации образов оказываются иногда полезными и для анализа изображений, получаемых с помощью зрительной системы. Распознать объект — это значит отнести его к одному из известных классов. Однако нужно подчеркнуть, что распознавание — это лишь одна из многих задач, стоящих перед зрительной системой. Исследователи, занимавшиеся распознаванием, разработали много простых методов получения характеристик по изображениям. Однако в этих методах изображения обычно рассматриваются как двумерные распределения яркости, а объекты не могут занимать произвольное положение.

В анализе сцен рассматривается задача перехода от простых описаний, полученных непосредственно по изображениям, к более сложным, представленным в форме, которая более приспособлена для выполнения определенного задания. Классической иллюстрацией этого служит интерпретация контурных рисунков (рис. 10.10). Здесь описание изображения совокупности многогранников дается в виде множества прямолинейных отрезков. Прежде чем его использовать, необходимо определить, какие области, ограниченные этими отрезками, относятся к одним и тем же объектам. Также желательно знать, как объекты расположены друг относительно друга. Подобным образом из простого описания можно получить сложное символическое описание изображения.

В анализе сцен символическое описание низкого уровня (контурные рисунки) используется для составления описания высокого уровня.

Последнее может содержать информацию о пространственных отношениях между объектами, об их форме и тождественности.

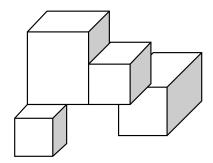


Рис. 10.10 - Символическое описание низкого уровня.

10.3.1 Обработка визуальной информации

Системы, решающие сложные и «интеллектуальные» задачи обработки и анализа видиоданных (изображений), которые обычно работают со входной информацией в виде полутоновых или многоспектральных изображений, объединяют под общим названием автоматизированных систем обработки изображений (АСОИз). Если рассматривается преобразование и обработка визуальной информации, реализованное в микрокомпьютерах, интегрально соединенных с фотоэлектронными матрицами интеллектуальных видеодатчиков, то такие системы называют системами машинного видения (СМВ).

Рассмотрим обработку визуальной информации, т. е. анализ и интерпретацию изображений [8, 9].

Описание объекта, полученное в результате предварительного анализа изображения, позволяет на этапе интерпретации отнести этот объект к определенному классу в принятой классификации. Этап интерпретации может быть реализован с помощью разнообразных способов, и рациональный выбор среди них наиболее подходящего зависит от назначения системы. Созданы различные способы анализа и интерпретации изображений. К таким способам в первую очередь могут быть отнесены: сравнение с эталоном, сравнение по характерным признакам образа, структурные способы и др. При этом способы анализа и интерпретации изображений существенно различаются в зависимости от того, к какому типу может быть отнесена данная схема: к двумерному, 2,5-мерному или трехмерному. Здесь необходимо отметить, что многие типы анализируемых сцен по своей сути являются двумерными. В последнее время достигнуты существенные успехи в области анализа двумерных объектов. С применением быстродействующих вычислительных средств параллельной структуры стало возможным создание различных СМВ, предназначенных для анализа и идентификации в реальном масштабе времени как бинарных, так и многотоновых двумерных изображений.

Алгоритмы, используемые в системах машинного видения, могут быть разделены на следующие классы [8,9]:

- Алгоритмы систем бинарного видения, действия которых сводятся к анализу черно-белых (бинарных) изображений.
- Алгоритмы многотонового видения, действие которых сводится к анализу изображений, содержащих много оттенков серого. Этот класс объединяет различные методы, начиная от обнаружения контуров и до корреляционного сравнения изображений по серой шкале.

- Алгоритмы активного трехмерного видения, основанные на использовании структурированного подсвета. Действие этого класса сводится к анализу изображений объектов, на которые спроектирован специальный регулярный световой рисунок. На основании анализа пространственной деформации подсвеченных световых структур представляется возможным делать заключение о пространственной форме объектов.
- Алгоритмы пассивного стереовидения, обрабатывающие совместно стереопару изображений анализируемой трехмерной сцены.
- Алгоритмы трехмерного видения, анализирующие дальностную картину, формируемую с помощью сканирующей лазерной дальномерной системы.
- Алгоритмы систем комплексного стерео- и дальнометрического видения и других СМВ.

При использовании алгоритмов бинарного видения предполагается, что все объекты, присутствующие на анализируемой сцене, в изображении могут появляться в виде двумерных пятен одного цвета (например, черного), в то время как фон может быть другого цвета. Это предположение используется для разделения изображения на пятна, соответствующие потенциальным объектам. Затем для определения набора характерных особенностей объектов, таких, как площадь, периметр и степень протяженности, система анализирует видимые пятна. В процессе обучения система запоминает особенности эталонных объектов и записывает их в память наряду с наименованием объекта. При сравнении особенностей предъявленного объекта с набором, записанным в памяти в период обучения, система может идентифицировать объект настолько достоверно, насколько уникальны (т. е. присущи только ему) его особенности.

Бинарные системы видения могут распознавать только те объекты, которые имеют отличительные двумерные формы. Чтобы обеспечить резкий контраст с фоном, в таких системах иногда используется специальное освещение объектов. Поскольку эти системы распознают объекты по характерным особенностям контурных очертаний, они обычно не могут идентифицировать объекты, которые касаются или перекрывают друг друга.

Объекты могут как частично перекрывать друг друга, так и иметь разный наклон по отношению к опорной плоскости. Одну из главных трудностей при анализе таких сцен представляет сегментация изображения. Сегментация проводится на основе выделения контурных линий на изображении и последующей интерпретации их как углов, окружностей и прямолинейных отрезков, описывающих форму деталей, расположенных в поле зрения системы. В результате затенения, наложения деталей и перспективных искажений могут исчезнуть некоторые признаки и возникнуть ложные признаки. СМВ должны быть способны работать с такими искаженными данными. Этого можно достичь за счет использования моделей в процедуре анализа, в которых содержатся априорные данные о геометрических и топологических свойствах контролируемых деталей. Каждая деталь описывается моделью, которую пользователь формирует в интерактивном режиме при обучении системы.

Задача анализа изображения состоит в формировании прототипа по элементарным признакам (примитивам), выделенным на этапе предварительной обработке

и подборе модели, наиболее близкой к этому прототипу. Как правило, процедура построения прототипа многостадийная: на каждом шаге с моделями сравниваются частичные наборы признаков и по результатам сравнения делается вывод о том, каким из имеющихся в списке признаков необходимо дополнить прототип на следующем шаге. Модель объекта включает в себя некоторые примитивы, а также может содержать дополнительную информацию типа уровня яркости, цвета или текстуры. При этом не обязательно описывать деталь полностью — достаточно указать несколько ее основных признаков.

Для идентификации многотоновых изображений используются различные методы. Идентификация таких объектов предполагает составление описания наблюдаемой картины, на основании которого затем проводятся сравнение и опознавание, причем для осуществления этого процесса необходимым является наличие определенных знаний относительно наблюдаемых объектов.

При обработке полутонового изображения прежде всего выделяются точки, для которых характерно резкое изменение уровня яркости. Затем путем соединения подобного рода точек формируется контурный рисунок и определяются значения линий контурного рисунка, для чего линии снабжаются обозначениями, указывающими на трансформацию поверхности по типу перехода к выпуклости, к впадине. Далее кадр разделяется на ряд однородных зон, и некоторые зоны объединяются в объекты. Затем путем сопоставления с моделями обеспечивается распознавание объектов. В общем случае результаты обработки сопоставляются с моделями некоторого числа объектов.

В стадии обучения благодаря предъявлению определенного числа прототипов объектов для каждого класса классификатор формируется таким образом, чтобы математическое ожидание для затрат на классификацию было минимальным и чтобы благодаря точному расчленению пространства признаков могло появляться только минимальное число команд возврата. В заключительной стадии анализа тест в соответствующих узлах двоичного дерева может продолжаться только в одном или другом направлении в зависимости от того, будет ли предложенное значение признака больше или меньше порогового значения, которое соответствует данному тестовому узлу. В завершение процесс достигает конечного узла дерева решений с указанием класса объекта или указанием для команды возврата.

Значительное развитие в последнее время получил класс СМВ, предназначенных для визуального анализа трехмерных сцен. Кроме пассивных стереоСМВ, действующих без использования специальных источников подсвета, созданы также активные анализаторы, в которых используются различные устройства специального структурированного подсвета. Структурированный подсвет, например щелевых, решетчатых и других узконаправленных источников света, широко применяется в СМВ, поскольку такое освещение позволяет простыми средствами анализировать трехмерную картину с достаточной точностью.

10.3.2 Система зрения роботов

Системы зрения промышленных роботов определяют следующие требования [4]:

- низкая стоимость оборудования и программного обеспечения;
- высокая скорость распознавания объектов, обеспечивающих синхронность

распознавания с технологическим процессом, в который интегрирована система зрения робота;

- высокая надежность распознавания объектов, их позиций и ориентации;
- высокая робастность к изменениям параметров объектов;
- малое количество распознаваемых объектов и их моделей (10–100);
- модели объектов могут быть заданы довольно точно на основе обучения или САПР;
- часть объекта имеет простую форму (прямые линии, плоские поверхности и т. д.);
- наличие локальных признаков (например, отверстие);
- освещение и фон можно настроить для упрощения распознавания.

Системы технического зрения роботов часто применяют в следующих производственных задачах:

- упаковка изделий;
- осмотр поверхностей (царапины, трещины, цвет), проверка размеров;
- распознавание, сортировка, схватывание и обработка объектов;
- чтение штрихового кода и текста;
- передвижение по дорогам, складам.

Для распознавания изображений используются 2D-методы с глобальными и локальными признаками. Рассмотрим пример промышленной системы зрения с методом распознавания изображения методом 2D, с использованием глобальных признаков.

Проблема состояла в том, чтобы распознать объекты, находящиеся в некотором рабочем пространстве либо на ленте конвейера. При этом объекты не соприкасались друг с другом и не накладывались один на другой. Кроме того, объекты были плоскими, а если какой-либо объект был трехмерный, то он мог находиться только в одном из трех устойчивых состояний. С учетом этих требований модель объекта была чрезвычайно проста: для каждого объекта и для каждого его устойчивого положения определялся вектор-прототип, состоящий из подходящих признаков, некоторые из которых указаны ниже:

- площадь;
- периметр;
- число отверстий;
- площадь отверстий;
- площадь самых маленьких описанных прямоугольников;
- полуоси самых маленьких описанных эллипсов;
- полуоси эллипса, который сопоставлен некоторой фигуре;
- самое маленькое и самое большое расстояние контура от центра тяжести.

Обучение. На этапе обучения используется множество изображений объектов каждом из возможных устойчивых положений (получаемых путем

смещения и вращения). При этом накапливается информация о среднем, дисперсии или других статистических характеристиках каждого из признаков. Используются те признаки, которые обладают лучшими дискриминационными характеристиками.

Разработка классификатора. Классификация выполняется одним из двух методов: «ближайших соседей» (расстояние до прототипа) или «двоичного дерева решений». Это зависит от стоимости определения значений признаков. Если затраты не велики, то применяется метод «ближайших соседей». Если затраты на прохождение отдельных признаков значительны, то использование иерархического дерева решений может быть выгоднее, так как в этом случае может оказаться, что некоторые признаки не требуются для классификации. Построение модели, разработка классификатора и обучение выполняются автономно. Поэтому не требуется, чтобы эти шаги выполнялись за минимальное время. После завершения указанных этапов программы выполняется распознавание в реальном времени. Время распознавания (или затраты на распознавание) играет важную роль.

Распознавание. Если анализируемая «сцена» содержит несколько объектов, то сначала выполняется разметка компонентов. Затем каждый из компонентов анализируется отдельно с помощью выделенного метода классификации.



Контрольные вопросы и задания к главе 10

- 1) Дайте определения терминам: «восприятие», «зрительное восприятие».
- 2) Опишите процесс моделирования зрительного восприятия реальности.
- 3) Проведите сравнение систем по глубине зрительного восприятия.
- 4) Определите понятие «машинное зрение».
- 5) Представьте основные сведения о распознавании образов.
- 6) Проанализируйте обобщенную схему системы распознавания.
- 7) С какими функциями Д. В. Гаскаров связывает распознавание?
- 8) Представьте и охарактеризуйте общую структуру системы распознавания и ее разработки.
- 9) Приведите задачи распознавания и классификации в редакции В. Н. Вагина.
- 10) Что может являться источником примеров для обучения?
- 11) Назовите значения признаков объектов.
- 12) Проведите сравнения объектов с использованием меры близости.
- 13) Определите меру близости объектов через расстояние.
- 14) Рассмотрите различные переходы к решению задачи об отнесении объекта к некоторому классу.
- 15) Сформулируйте задачу обучения без учителя.
- 16) Опишите алгоритм, основанный на понятии порогового расстояния.

- 17) Приведите примеры и решите задачу по алгоритму порогового расстояния.
- 18) Запишите и объясните формулу Байеса в методе распознавания объектов.
- 19) Представьте Байесовское решающее правило.
- 20) Какой подход базируется на структурных отношениях образа?
- 21) Охарактеризуйте синтаксический метод распознавания и приведите пример задачи распознавания изображений и, в частности, анализ сцен.
- 22) Представьте и опишите системы синтаксического распознавания образов.
- 23) Сформулируйте грамматики, отличающиеся типом правил подстановки.
- 24) Рассмотрите и опишите пример контекстно-зависимой грамматики.
- 25) Перечислите типичные задачи компьютерного зрения.
- 26) Сформулируйте основные функции компьютерного зрения.
- 27) Расскажите и постройте структурную схему системы SIGMA.
- 28) Перечислите и опишите этапы понимания изображения.
- 29) Опишите базовые операции при анализе изображений.
- 30) Проведите анализ обработки изображений, классификации образов и сцен в машинном зрении.
- 31) Дайте характеристику и перечислите классы алгоритмов в системах машинного видения.
- 32) Перечислите требования, определяющие систему зрения промышленных роботов.
- 33) Приведите производственные задачи, которые часто применяются в системах технического зрения роботов.
- 34) Какие методы используются для распознавания изображения в системах зрения роботов?
- 35) Перечислите этапы распознавания образов в техническом зрении.



Литература к главе 10

- [1] Александров В. В. Представление и обработка изображений. Рекурсивный подход / В. В. Александров, Н. Д. Горский. М., 1989. 190 с.
- [2] Бонгард М. М. Проблема узнавания / М. М. Бонгард. М: Наука, 1967. 320 с.
- [3] Бондарев В. Н. Искусственный интеллект : учеб. пособие для вузов / В. Н. Бондарев, Ф. Г. Аде Севастополь: Изд-во СевНТУ, 2002. 615 с.
- [4] Достоверный и правдоподобный вывод в интеллектуальных системах / В. Н. Вагин [и др.]; под ред. В. Н. Вагина и Д. А. Поспелова. М.: Физматлит, 2004. 704 с.
- [5] Гаскаров Д. В. Интеллектуальные информационные системы: учебник для вузов / Д. В. Гаскаров. М. : Высшая школа, 2003. 413 с.
- [6] Дуда Р. Распознавание образов и анализ сцен / Р. Дуда, П. Харт. М. : Мир, 2001.
- [7] Катыс Г. П. Системы машинного видения : анализ состояния и перспективы развития / Г. П. Катыс, П. Г. Катыс // Приборы и системы управления. 1999. №9. С. 57–67.
- [8] Катыс Г. П. Восприятие и анализ оптической информации автоматической системой / Г. П. Катыс. М. : Машиностроение, 1986.-415 с.
- [9] Попов Э. В. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта / Э. В. Попов, Г. Р. Фридман. М. : Изд-во «Наука», 1976.-456 с.
- [10] Искусственный интеллект : в 3 кн. / под ред. Д. А. Поспелова. М. : Радио и связь, 1990. Кн. 2: Модели и методы.
- [11] Форсайт Д. А. Компьютерное зрение. Современный подход: пер. с англ. / Д. А. Форсайт, Ж. Понс. М. : Издательский дом «Вильямс», 2004. 928 с.
- [12] Фор А. Восприятие и распознавание образов : пер. с франц. А. В. Серединского / А. Фор ; под ред. Г. П. Катыса М. : Машиностроение, 1989.-236 с.

ЗАКЛЮЧЕНИЕ

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

В десяти разделах учебного пособия излагаются основные вопросы искусственного интеллекта: структура исследований, задачи и методы их исследования, основные виды логического вывода, неопределенность знаний, модели знаний, способы обработки знаний, системы планирования, понимание естественного языка, машинного перевода, зрительного восприятия мира и, наконец, экспертные системы (интегрированные и гибридные).

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

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

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

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

- разработка и автоматизация технологий извлечения знаний и формирования баз знаний;
- разработка интеллектуальных человеко-машинных интерфейсов;
- вложение семантической информации в формализмы представления знаний, накопление и использование их;
- организация текущих знаний в самостоятельные, специализированные

186 Заключение

фрагменты для практического использования;

- структурирование статических и динамических знаний;
- адаптация и эффективность решения задач, в основанных на знаниях системах (планирование, понимание).

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

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

ГЛОССАРИЙ

Абдуктивный вывод — схема: даны правило и результат, требуется вывести частный случай — причину. Если правило выражается через связь — импликация (не обязательно), то правило силлогизма можно представить следующим образом:

$$\frac{A \to C, C}{A}$$
,

где A может быть причиной C или доводом в пользу того, чтобы C было истинным.

Aбдукция — процесс формирования объясняющей гипотезы. Точнее, при заданной теории и наблюдении, предложенном для объяснения, абдуктивный вывод должен определить одно или более наилучших объяснений наблюдения на основе заданной теории.

Aгрегатный объект — составной объект, образованный из других объектов, которые рассматриваются как его составные части.

Байесовский метод распознавания объектов — построение решающих правил на основе статистических свойств классов.

Ведение диалога — определение его структуры и ранга роли, которую система и пользователь выполняют на текущем шаге диалога.

Восприятие зрительной информации—процесс извлечения из потока лучевой энергии, попадающего на рецепторы зрительной системы (искусственной или естественной), информации об окружающей среде.

Восприятие в искусственном интеллекте—процесс преобразования информации о внешнем мире, поступающей от анализаторной системы в форму для ее использования в интеллектуальной системе.

Геометрический метод распознавания объектов — построение решающего правила по критерию расстояния (меры близости), объекта к классу.

Децентрализованный процесс — разработка коллективного поведения большого числа взаимодействующих между собой интеллектуальных агентов. При этом интеллект агента рассматривается как подсистема управления деятельностью в процессе взаимодействия с другими агентами.

Диагностика — процесс соотнесения объекта с некоторым классом объектов и/или обнаружение неисправностей в системе (отклонение параметров системы от нормативных в технике и в живых организмах).

Диалог — процесс достижения его участниками определенных согласованных целей путем обмена связанными высказываниями, выраженными в языке о некотором реальном или гипотетическом мире(проблемной области).

Дискурс — связный текст высказывания участников общения.

Генерация высказывания — формирование выходных высказываний на ЕЯ.

Знания — хорошо структурированные данные или данные о данных (метаданные).

Индивидный объект семантической сети — выделенный каким-то образом единичный представитель (экземпляр) класса.

 $\mathit{Индукция}$ — метод опытного назначения, путь движения мысли от единичного к общему понятиям и утверждениям.

 $\mathit{Индукция}$ — вывод, специфический тип рассуждений (формальная, материальная индукция).

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

Интегрированные экспертные системы — программная среда, в архитектуре которой наряду с традиционным компонентом ЭС, содержатся некоторые компоненты N, расширяющие функциональные компоненты ЭС (БД, пакеты прикладных программ, обучающие системы и т. д.).

Интенсионал понятия — смысл, который мы вкладываем в данное понятие, т. е. интенсионал характеризует концепт данного понятия, его содержание.

Интеллектуализированная система — это информационно-вычислительная система с интеллектуальной поддержкой при решении задач с участием ЛПР.

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

Интерпретация — процесс определения смысла данных (построение описаний по наблюдаемым данным).

Искусственный интеллект — наука, ставящая своей целью создание искусственных систем, достаточно хорошо имитирующих интеллектуальную деятельность человеческой личности.

Кластеризация — отнесение объекта к неизвестному классификатору класса, т. е. процесс выделения новых классов.

Komnью mepная лингвистика — широкая область использования компью терных инструментов — программ, компью терных технологий организации и обработки данных — для моделирования в тех или иных условиях.

Погический вывод в математической логики — частный случай рассуждений, когда множество аргументов фиксировано, нетривиальные средства (например,

проверка на непротиворечивость) не используются и применяются только правила достоверного вывода, по которым из истинных аргументов (посылок) можно получить лишь истинные заключения.

Машинное зрение — направление исследований, ставящее своей целью моделирование зрительной информации восприятия, а в процесс функционирования накапливает образные и логические знания.

 $\it Mauuuhhbi u nepe \it Bod - uc nonsobahue \it BM определенных и постоянных для данного вида материала соответствий между словами и грамматическими явлениями разных языков.$

Метод ключевых состояний и ключевых операторов—это сведение задачи к подзадачам, позволяющие использовать наиболее вероятные ключевые состояния и ключевые операторы при решении задачи.

Мониторинг — непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы.

Монотонность — свойства логических выводов. Для полного набора знаний справедливость полученных выводов не нарушается с добавлением новых фактов, т. е. если $A_1, A_2, \ldots, A_n \vdash B_1$, то $A_1, A_2, \ldots, A_n, \{F\} \vdash B_1$, где $\{F\}$ — множество добавочных утверждений не отменяет ранее выведенного утверждения B_1 .

 $\it Headeкватность описания \, \it задачи - p \rm p \rm g$ элементов задачи временно описанных лишь по аналогии.

Неизвестность описания задачи — практическое отсутствие информации.

Недоопределенность описания задачи — для некоторых элементов определены не их точные описания, а лишь множество, которым эти описания принадлежат.

Недостаточность описания задачи — собрана не вся необходимая информация.

Неоднозначность описания задачи — вся возможная информация собрана, но полностью определенное описание не получено и не может быть получено.

Неполнота описания задачи — собрана еще не вся возможная информация.

Непрерывное планирование — планирование в течении всего срока существования плана.

Непротиворечивость — свойство, которое сводится к тому, что исходная система аксиом и правил вывода не должна давать возможность выводить формулы, не принадлежащие заданному множеству формул с выбранными свойствами.

Неточность — относится к содержанию информации (или значению сущности) и наряду с неполнотой и противоречивостью должна обязательно учитываться при представлении знаний в БЗ. Возникновение неточных знаний связанно с:

- ошибками в данных;
- ошибками в правилах вывода;
- использованием неточных моделей.

Обобщенный объект семантической сети— некоторая собирательная абстракция реально существующего объекта, процесса или явления предметной области.

Обучение — диагностика, интерпретация, планирование, проектирование. Системы обучения выполняют такие функции, как диагностика ошибок,

подсказывание правильных решений, аккумулирование знаний о гипотетическом «ученике» и его характерных ошибках, диагностирование слабости в познаниях обучаемых и нахождение соответствующих средств, для их ликвидации. Системы обучения способны планировать акт общения с учеником.

Общение — коммуникативное взаимодействие (в данном случае) человека с компьютером.

Обработка высказываний — формирование или определение заданий на решение задач или подзадач на данном шаге диалога.

Образец в базе знаний — некоторая информационная структура, определяющая обобщенную информацию (условие, состояние и т. д.) окружающей действительности, при которой активизируется правило.

План — целенаправленная система поиска решения задачи.

Поиск по образцу — циклический процесс вывода новых знаний.

Полнота E3— множества формул с заданными свойствами исходная, где система аксиом и правил вывода должны обеспечить вывод всех формул, входящих в это множество.

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

Планирование без использования датичков — метод предусматривающий создание стандартных, последовательных планов, которые должны выполнятся без учета результатов восприятия.

Понимание — преобразование поступающих от пользователя высказываний на естественном языке в высказывания на языке внутреннего представления.

Представления знаний — соглашение о том, как описывать реальный мир.

Понимание текста интеллектуальной системой— ответы на все прямые вопросы и представление ситуации реального мира по тексту. Это три уровня интерпретации: морфологический анализ и предсинтез, синтаксический анализ, семантический анализ.

Принципы дедуктивного вывода для высказывания:

- Modus Ponendo Ponens: «Если истинна импликация $A \to B$ и A истинно, то B истинно».
- Modus Tollendo Tollens: «Если истинна импликация $A \to B$ и B ложно, то A ложно».
- Modus Ponendo Tollens: «Если A истинно и конъюнкция $A \wedge B$ имеет результатом ложь, то B ложно».

• Modus Tollendo Ponens: «Если A ложно и дизьюнкция $A \lor B$ истинна, то B является истиной».

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

Продукционная система — программная система, которая может быть представлена состоящей из трех модулей, или блоков:

- глобальная база данных (просто база, или Б-Модуль);
- множество правил-продукций (П-Модуль);
- система управления или интерпретатора (У-Модуль).

Продукция — правило-продукция, представляющая собой пару: ситуация → действие, посылки → заключение, причина → следствие и т. п.

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

Процесс дедукции основан на законе тождества, где частное подводится под общее на том основании, что оно по содержанию тождественно с общим; то же самое положение можно заметить и в заключении от частного к частному.

Разрешение конфликтов — управление порядком применения правил в конфликтном множестве.

Распознавание — это идентификация объекта в качестве элементов известного классификатору класса.

Рассуждение — построение последовательности аргументов, вынуждающих принятие некоторого утверждения, которое и является целью рассуждения.

Связь между понятиями семантической сети — минимальный объем знаний, простейший факт, относящийся к двум понятиям. Более сложные утверждения в рамках семантической сети могут быть определены путем выделения соответствующих подграфов.

Семантическая сеть — система знаний некоторой предметной области, имеющую определенный смысл в виде целостного образа сети, узлы которой соответствуют понятиям и объектам, а дуги — отношениям между объектами.

Семантический анализ — построение семантического графа предложения для выявления смысла предложения.

Синтаксический анализ — грамматический разбор предложений, это построение синтаксической структуры в виде дерева или даже графа.

Синтаксический метод распознавания—выделение не признаков, а простейших структурных элементов и получение сложной иерархической структуры объекта.

Система искусственного интеллекта— это компьютерная, креативная система (многофункциональная, интегрированная, интеллектуальная) со сложной структурой,

использующая накопление и корректировку знаний (синтаксической, семантической, прагматической информации) для постановки и достижения цели (целенаправленного поведения), адаптации к изменениям среды и внутреннего состояния путем изменения среды или внутреннего состояния.

Системы когнитивной графики — совместная работа двух систем переработки информации: зрительной, создающей образную картину мира, и символической, способной к абстрактному мышлению, к оперированию с понятиями, интегрирующими образы внешнего мира. Возможность перехода от зрительной картины к ее текстовому (символическому) описанию и от текста к некоторой зрительной картине, составляют, по-видимому, основу того, что называется мышлением.

Сопоставление — основной механизм вывода из фреймовой модели. Различают сопоставления: синтаксическое, семантическое, параметрическое.

Статистический машинный перевод — технология перевода, основанная на статистическом анализе (т. е. сравнении больших объемов языковых форм).

Стоимость дерева решения— суммарная стоимость, представляющую собой суму стоимости всех дуг в дереве решения и максимальная стоимость, равная стоимости пути между двумя вершинами дерева решения.

Cуждение — это законченная мысль, которую, можно выразить на естественном языке в одной из следующих четырех форм:

- 1) все X являются $P (\forall X) P(X)$;
- 2) никакой X не является $P (\forall X) \sim P(X)$;
- 3) некоторые из X являются $P (\exists X) P(X)$;
- 4) некоторые из X не являются $P (\exists X) \sim P(X)$.

Cиенарий — это структурированное представление, описывающие стереотипную последовательность в частом контексте.

Техническое зрение — система, имеющая узкоспециализированное назначение и содержащая минимально необходимый набор датчиков информации и средств обработки для выполнения заданных функций.

Типовая архитектура ЭС состоит из: интерфейса, ядра ЭС, базы знаний, решателя, подсистемы объяснений и подсистемы приобретения знаний.

Управление выводом — прямая или обратная цепочка рассуждений и механизм разрешения конфликта в продукционной системе.

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

Условное планирование — действия в условиях ограниченной недетерминированности по созданию условного плана с различными ответвлениями для разных ситуаций.

 Φ акторизованное пространство — непересекающиеся подпространства (классы) с частичными (неполными) решениями.

 Φ изическая неопределенность — описание неопределенности объектов реального мира с точки зрения наблюдателя.

Формальная система в логике задается четверкой:

$$M = \{T, P, A, R\},\,$$

где T — множество базовых элементов; P — множество синтаксических правил; A — множество аксиом; R — множество правил вывода.

 $\Phi pe \check{u}_{M} - \varphi pa \Gamma ment$ сети, объединяющий сущности на основе семантической близости.

Фундаментальные исследования—процессе разработки новых моделей и методов для решения задач, считающихся интеллектуальными и не поддававшихся ранее формализации и автоматизации; исследования, связанные с новыми идеями решения задач на ЭВМ, с разработкой новых технологии программирования и переходом к компьютерам не фон-неймановской архитектуры.

Экстенсионал понятия — совокупность всех его допустимых денотатов, соответствующих концепту этого понятия.

Учебное издание

Павлов Станислав Николаевич СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ЧАСТЬ 2

Учебное пособие

Корректор Осипова Е. А. Компьютерная верстка Лигай Т. А.

Подписано в печать 26.09.11. Формат 60x84/8. Усл. печ. л. 22,78. Тираж 200 экз. Заказ

Издано в ООО «Эль Контент» 634029, г. Томск, ул. Кузнецова д. 11 оф. 17 Отпечатано в Томском государственном университете систем управления и радиоэлектроники. 634050, г. Томск, пр. Ленина, 40 Тел. (3822) 533018.