

III. ОБУЧЕНИЕ ЦЕЛЕСООБРАЗНОМУ ПОВЕДЕНИЮ

М. М. Бонгард, И. С. Лосев, М. С. Смирнов

ПРОЕКТ МОДЕЛИ ОРГАНИЗАЦИИ ПОВЕДЕНИЯ — «ЖИВОТНОЕ»

Общая цель моделирования. До сих пор наша группа в своих попытках понять и промоделировать мыслительные процессы ограничивалась «проблемой узнавания». Точнее, придумывались алгоритмы нахождения решающего правила для нужной классификации внешних объектов. При этом описания предметов, содержащие информацию, достаточную для определения класса, уже были составлены в удобной для логического анализа форме, а на описания объектов из материала обучения наклеены ярлыки классов и, тем самым, сказано, какие классы сейчас важны. Благодаря усилиям многих исследователей теперь удастся неплохо справляться с задачами такого типа.

В жизни, однако, задача узнавания встречается в гораздо более сложной форме. Часто нет знания о том, какое именно описание достаточно для нахождения решающего правила. Более того, часто неясно, на какие классы следует разбивать множество объектов, подлежащих узнаванию.

Кроме классификации внешних объектов мышление человека и животных решает задачи и совсем другой природы. Приходится, например, рассматривать и классифицировать некоторые внутренние объекты, определять, годится ли данное возможное действие для решения стоящей сейчас задачи. Но еще прежде, чем выбирать действие, лучшее из нескольких вспомненных или придуманных, необходимо из множества актуальных задач выбрать именно ту, которой лучше заняться теперь же. Само множество актуальных задач тоже синтезируется внутри живой системы.

Таким образом, если нас интересует поведение в целом, то возникает задача не только создать модели отдельных мыслительных блоков, таких, как механизм узнавания или модель ассоциативной памяти, но и построить общую схему всей организации поведения. Это нужно не столько для получения результатов, непосред-

ственно полезных в практических задачах, сколько для понимания того, как мыслим мы сами. Поэтому мы предприняли попытку построить модель, воспроизводящую в грубых чертах поведение человека или хотя бы «разумных» животных.

Сначала кратко опишем проект животное в целом, а далее более подробно остановимся на некоторых конкретных вопросах: выборе действия, способах переключения внимания, устройстве памяти и, наконец, многоступенчатом поиске действия и прогнозе.

1. Общее описание. 1.1. Мир животного. Мир, как и само животное, имеет смысл реализовать внутри вычислительной машины. Простейшие варианты удобно отлаживать, предлагая модели задачи на построение и организацию сложных последовательностей элементарных действий. Моделирование мира внутри ЭВМ позволяет, при желании, избавиться от необходимости решать проблемы динамики отдельного движения и сосредоточить усилия на проблеме выбора нужной последовательности действий, а не на трудностях осуществления элементарных действий. Мы выбрали следующий простейший вариант мира и «облика» животного, о котором удобно думать на первых этапах моделирования.

Мир — это кусок плоскости, разбитый на клетки. Внутри клеток находятся предметы. В мире действуют законы, которые можно поначалу выбрать весьма простыми и похожими на законы реального мира.

1.2. Органы чувств и эффекторы. Животное обладает набором рецепторов, как внешних (таких, как зрение), так и внутренних (например, чувство, дающее координаты «руки» или «ноги» относительно «тела»). Все, что в данный момент времени животное может непосредственно воспринимать во внешнем мире, определяется заданным при конструировании набором рецепторов.

У животного есть и такие эффекторы, как нога (оно способно передвигаться), рука (ею можно перемещать предметы) и т. д. Фактически в нашей модели эффекторы — это исполнительные программы, изменяющие, после передачи им управления, окружающую среду и состояние животного в соответствии с заданными законами природы.

1.3. Описание ситуации. Совокупность показаний рецепторов на некоторый момент будем называть *первичным описанием ситуации*. Это сырой материал. Результаты его обработки называются *обобщенными описаниями ситуаций*. В них могут входить не только показания рецепторов, но и утверждения о соотношениях между этими показаниями. Типы возможных утверждений не перечисляются явно при конструировании системы, а задаются набором элементарных предикатов и элементарных функций от показаний рецепторов, а также алгоритмами, которые из этих элементов могут конструировать то или иное сложное утверждение, в зависимости от имеющегося опыта. Обобщенные описания обыч-

но соответствуют не одной внешней ситуации, а целому их множеству.

1.4. Задачи и потребности. Интуитивное представление о том, что такое «задача», есть у всех. Формально, в животном задача определяется описанием (обычно обобщенным) некоторой целевой ситуации Z . Процесс решения задачи — это планирование и выполнение действий, предположительно годных для того, чтобы была достигнута ситуация Z . Задача считается решенной, если показания органов чувств становятся такими, что утверждения о показаниях рецепторов, перечисленные в Z , выполняются.

Откуда берутся задачи? В живых существах есть врожденные или приобретенные в течение жизни *потребности*. Их можно рассматривать как внутренние, постоянно действующие генераторы задач. В нашем животном мы также предусматриваем подобный генератор. Это — механизм, изменяющий состояние некоторого счетчика; как только показания счетчика превысят порог, возникает «желание» — перед мозгом животного встает задача ввести показания счетчика в заданные пределы (подробнее о механизмах обращения внимания на неудовлетворенные потребности см. п. 2.3). Когда это удастся, желание исчезает. Нужные для этого действия определяются законами мира и конструкцией животного. Например, можно сделать так, чтобы показания счетчика росли с течением времени и уменьшались до нуля только в случае, если животное найдет в мире предмет определенного типа и положит его в определенное место. На человеческом языке такая потребность аналогична голоду, который удовлетворяется с помощью нахождения и поедания пищи. Упомянутый счетчик в животном мы назовем рецептором голода G , так что задача, постоянно возникающая у животного, имеет обозначение $Z: G < G_0$.

У человека имеется много потребностей, причем значительная их часть возникает в течение жизни. У животного также предполагается возникновение новых «потребностей», управляемых законами, обусловленными конструкцией и предысторией: каждая задача, не слишком легкая, но много раз в различных ситуациях решавшаяся (в процессе удовлетворения уже имеющихся потребностей), сама переводится в ранг потребности. Показания счетчика такой потребности обращаются в нуль только в результате решения соответствующей задачи.

1.5. Место обобщения в схеме. Выбор действий, нужных для решения стоящей задачи, может быть основан как на априорных алгоритмах, вложенных конструкторами, так и на использовании накопленного животным опыта. Нас интересует создание модели, которая в течение жизни приобретает большой объем знаний, полезных для решения встающих перед ней задач. Очевидно, в такой модели память обученной системы будет хранить записи трех видов: сырые наблюдения, результаты их обобщений — широко применимые утверждения о достаточных условиях наступления интересующих систему событий, и «априорные знания», заложен-

ные конструкторами при создании системы. Отдельные записи, являющиеся единицами памяти, мы будем называть *фактами*. (Хранение априорных и накопленных знаний на равных правах и в одинаковой форме позволит системе исправлять неверные априорные представления о свойствах мира и целесообразных действиях.)

Таким образом, при эволюции модели должно появляться все большее число обобщений, описывающих, какие действия надо применять, находясь в том или ином классе ситуаций, для решения тех или иных задач. Некоторые операции обобщения и фрагменты языка записи ситуаций, с которыми работают эти операции, приведены в [1, 2]. Здесь мы просто допустим, что животное обладает механизмом обобщения и фиксированным списком «задач обобщения». (Наиболее простая из таких задач, задача обобщения начальных ситуаций, разобрана в [2].) Как только возникает необходимость в решении одной из них, механизм обобщения включается и совершает свою работу параллельно с попытками животного найти нужное действие на основе старого запаса сведений и обобщений.

1.6. Блочная структура модели. Животное состоит из нескольких однотипных по своим внутренним механизмам блоков. Каждый блок предназначен для решения совокупности задач определенного рода. В соответствии с кругом задач, он снабжается подходящим набором рецепторов и исполнительными органами. Ему также даются определенные возможности взаимодействия с другими блоками. В память блока заносятся факты (форму записи факта см. 3), относящиеся к его действиям и к воспринимаемым его рецепторами сторонам ситуации.

Работа блока заключается в том, что он по данной задаче и текущей ситуации, пользуясь своей памятью, находит и выполняет действие. *Действие блока* — это либо команда исполнительной программе, совершающей заданную операцию «без размышлений», либо постановка задачи другому блоку, либо постановка новой задачи самому себе.

Так, например, блок «рука» (блок, отдающий приказы эффектору «рука») может дать команду «передвинуть руку налево» исполнительной программе или же поставить задачу блоку «нога» «приблизить животное к предмету», если блок рука хочет его взять, или поставить задачу себе — изготовить приспособление для добычи предметов, не доступных руке без орудия.

Значительная часть действий блока, не сопровождаясь внешними изменениями, будет приводить лишь к переформулировке задачи и выбору для дальнейшего использования другой части общей памяти системы — памяти того блока, которому отдано управление. В таких случаях блок будет учиться формулировать законы, описывающие, как от ситуации и задачи зависит выбор подчиненных блоков и постановка им задач. При этом окажется, что, решая свои задачи, блок будет обрабатывать не все записи,

имеющиеся в памяти системы, а лишь небольшую долю записей, имеющих «прямое отношение к делу». Такое построение дает надежду преодолеть одну из трудностей «интеллектуальных задач» — слишком большое количество «начальных записей», комбинации которых необходимо перебирать¹.

С этой точки зрения идея блочной системы эквивалентна предложению о построении многоэтажных узнающих систем, формирующих признаки полезности применения признаков младшего этажа [3]. Однако блочность приводит к более единообразной системе и обладает рядом других полезных свойств.

Например, в памяти каждого блока факты будут содержать показания лишь небольшой части набора рецепторов всей системы, той части, которая предположительно существенна для решения задач блока. С такими, как бы предварительно обобщенными, короткими записями алгоритмам обобщения работать намного легче.

1.7. Понятие «внимание». Каждый блок обладает, кроме основной памяти, еще специальным рабочим полем, которое мы будем называть *вниманием*. Перед использованием факта, находящегося в памяти, его переносят на это поле. Выбор из памяти во внимание производится некоторым автоматом, способным работать в нескольких режимах. Блок, решая задачу, задает автомату режим поиска, определяющий условия, которым должны удовлетворять выносимые во внимание факты. Такими условиями могут быть: сходство ситуации, упомянутой в факте, с текущей, тождество действия в факте и действия, находящегося во внимании и т. д. Кроме вспомненных фактов во внимании находятся результаты их обработки, а также два поля — Z_0 и S_0 , на одном из которых (Z_0) — описание решаемой в данный момент задачи, а на втором (S_0) — описание текущей ситуации — совокупность показаний рецепторов блока на данный момент.

Предположим, что для решения некоторой задачи блок выбрал действие, состоящее в постановке себе промежуточной задачи. Выполнение этого действия приводит к смене текущей задачи блока. Во внимание из памяти извлекаются относящиеся к новой задаче факты, а факты, заложенные там вместе с первой задачей, исчезают. Что происходит с этим накопленным ранее содержанием внимания? Его стирание неоптимально, поскольку в нашем обычно медленно меняющемся мире факт, который был недавно извлечен из памяти, на который было «обращено внимание», с высокой вероятностью потребуетя вскоре еще раз.

Поэтому более эффективное использование уже найденной в памяти информации будет обеспечиваться следующим правилом: при постановке задачи Z самому себе блок A создает свою копию A' и передает ей управление, ставя перед ней задачу Z . При этом

¹ Для программы, доказывающей теоремы, «начальная запись» — это аксиома или уже доказанная теорема; для программы, обучающейся распознаванию, — это элементарный признак и т. д.

долговременная память у A и A' общая. В начальный момент работы блока A' его внимание то же, что было у A , — только задача иная. Затем внимание A' загружается новыми фактами, а не относящиеся к Z факты, находящиеся в нем, выбрасываются. Внимание же блока A остается нетронутым, так что, если возможна параллельная работа блоков A и A' , она будет производиться; если же она невозможна, то блок A будет бездействовать до передачи ему управления, после чего он сможет возобновить работу, не теряя времени на поиск в памяти, если только ситуация не слишком изменилась.

1.8. Выбор действия и ценности. Выбор действия блоком содержит три этапа: предварительный поиск подходящих действий, прогноз результатов каждого из них и, наконец, окончательный выбор. На первом этапе по заданной текущей ситуации S_0 и текущей задаче Z_0 блок ищет в памяти факты¹, применимые к текущей ситуации и содержащие описания таких действий, в результате которых бывала решена какая-то часть текущей задачи. Каждому такому факту жесткий механизм приписывает число — его *ценность* по отношению к данной задаче в данной ситуации. Эта ценность зависит от того, 1) насколько факт применим в текущей ситуации, 2) сколь большую долю текущей задачи он обещает решить²; 3) каково отношение числа тех применений факта, в которых описанный в нем результат действия действительно достигался, к общему числу его применений³.

В простейшем случае в факте указано рекомендуемое им действие. Тогда можно считать, что ценность этого действия совпадает с ценностью факта. Если таких фактов найдено несколько, то во внимании окажется набор альтернативных действий, имеющих различные ценности.

Может быть и так, что в памяти не будет найдено фактов с явными и полными описаниями подходящих действий, но такие описания удастся получить путем сопоставления нескольких обобщенных фактов.

Так или иначе в результате обработки, происходящей во внимании, появится набор альтернативных действий, снабженных ценностями, значения которых зависят от ценностей фактов и способа их обработки. После того, как такой набор получен, блок производит случайный выбор одного из действий, причем вероятность выбора каждого действия растет вместе с его ценностью.

Казалось бы, такая схема, где выбор действия совершается в соответствии с ценностями, определяемыми жестким механизмом, должна приводить

¹ Т. е. задает автомату памяти соответствующий режим поиска. Алгоритм поиска (устройство автомата) описан в 3.

² Смысл слова «обещает» будет уточнен в п. 3.3.

³ Заметим, что эта статистика зависит от степени обобщения оцениваемого факта. Следовательно, ценность факта будет зависеть как от внешнего опыта, так и от проделанной «внутренней работы» по отысканию полезных обобщений.

к необучаемому негибкому поведению. Однако это не так, поскольку при таком подходе обучаемость вкладывается в алгоритмы обобщения, которые должны строить факты, имеющие высокую ценность и дающие возможность верно найти действие. Если считать, что применимость факта к данной ситуации измеряется расстоянием от условий его применимости до ситуации (см. [1]), то операции обобщения, описанные в [2], направлены именно на порождение фактов с большой ценностью.

После выбора действия на первом этапе блок переходит ко второму этапу. Теперь в памяти идет поиск фактов, применимых к текущей ситуации и содержащих описания результатов выполнения действий, близких к выбранному действию D . Ценность таких фактов снова зависит от трех обстоятельств, приведенных выше, однако пункт 2) заменяется на 2') насколько действие факта близко к действию D .

После завершения поиска во внимании оказываются дополнительные факты, содержащие прогнозы результатов выполнения D , со своими ценностями. Теперь ценности действий, построенных на первом этапе, перевычисляются с учетом этой новой информации.

Если пересчет не снизил существенно ценность действия D , то оно выполняется. В противном случае опять выбирают какое-нибудь из альтернативных действий, пользуясь уже их новыми оценками. Описанная деятельность может продолжаться с многократными возвратами от прогноза к поиску действий и обратно, заканчиваясь каким-то окончательным выбором.

Нам кажется, что у человека имеются механизмы, запрещающие при некоторых условиях (перерасход времени, изменение ситуации) продолжение работы по вычислению ценностей или поиску фактов и требующие немедленного совершения действия или перехода к другой задаче. У животного также должны быть аналогичные возможности. Некоторые устройства, обеспечивающие такое поведение, будут описаны в следующем параграфе.

2. Переключение внимания

Блочная система особенно удобна, когда поиск действия может вестись одновременно в нескольких блоках. Однако маловероятно, что удастся запараллелить работу всей системы. Даже у человека, мозг которого в высокой степени приспособлен к параллельной работе, существуют уровни, устроенные явно последовательно. Самонаблюдение показывает, что так называемые «сознательные операции», или действия, проходящие через внимание человека, как правило, выполняются последовательным механизмом.

В параллельно-последовательной системе работа, порученная последовательному механизму, часто оказывается «узким местом», в котором особенно важно, какие задачи и в каком порядке решаются. С этой точки зрения хотелось бы организовать взаимодействия блоков внутри системы в предположении, что в каждый момент имеется лишь один *активный блок*: во внимании лишь одного блока появляются новые факты, взятые из памяти или полученные в результате обработки уже имеющихся во внимании записей. Некоторые изменения могут происходить и в «не работающих» блоках, как это будет описано ниже.

Итак, предположим, что в каждый момент активно работает лишь один блок системы. Внимание этого блока мы будем называть *вниманием системы*¹. Совокупность же вниманий всех блоков, имеющих нерешенные задачи, назовем *краевым сознанием*. В похожем смысле этот термин применял Адамар [4] и ранее Джемс [5]. Действие блока может заключаться в *переключении внимания* — постановке задачи другому блоку и передаче ему управления.

Если блок *A* поставил задачу блоку *B*, то блок *A* мы назовем *начальником*, а блок *B* — *подчиненным*. (Для другой задачи они могут поменяться местами.) После того, как подчиненный решил задачу, он возвращает управление начальнику. Во внимании каждого блока хранится имя его начальника и, если блок совершил действие, имя его подчиненного. Таким образом, краевое сознание организовано в каждый момент в некоторое дерево — *дерево задач*, типа представленного на рис. 1. Узлами этого дерева являются внимания блоков. (Некоторые из узлов могут быть вниманиями блоков-двойников, образовавшихся, например, при постановке блоком подзадачи самому себе.)

На рис. 1 каждый прямоугольник и кружок над ним изображают внимание блока. Внутри кружка — имя задачи блока. Номера внутри прямоугольников обозначают альтернативные действия, которые блок оценил как перспективные для решения своей задачи. Одно из них блок выполняет, т. е. ставит соответствующую задачу подчиненному.

2.1. Механизм переключения внимания. Работа внутри внимания системы приводит к переключению внимания, которое сопровождается либо ростом дерева подзадач из-за постановки новой задачи, либо исчезновением ветви дерева из-за решения или отмены одной из задач.

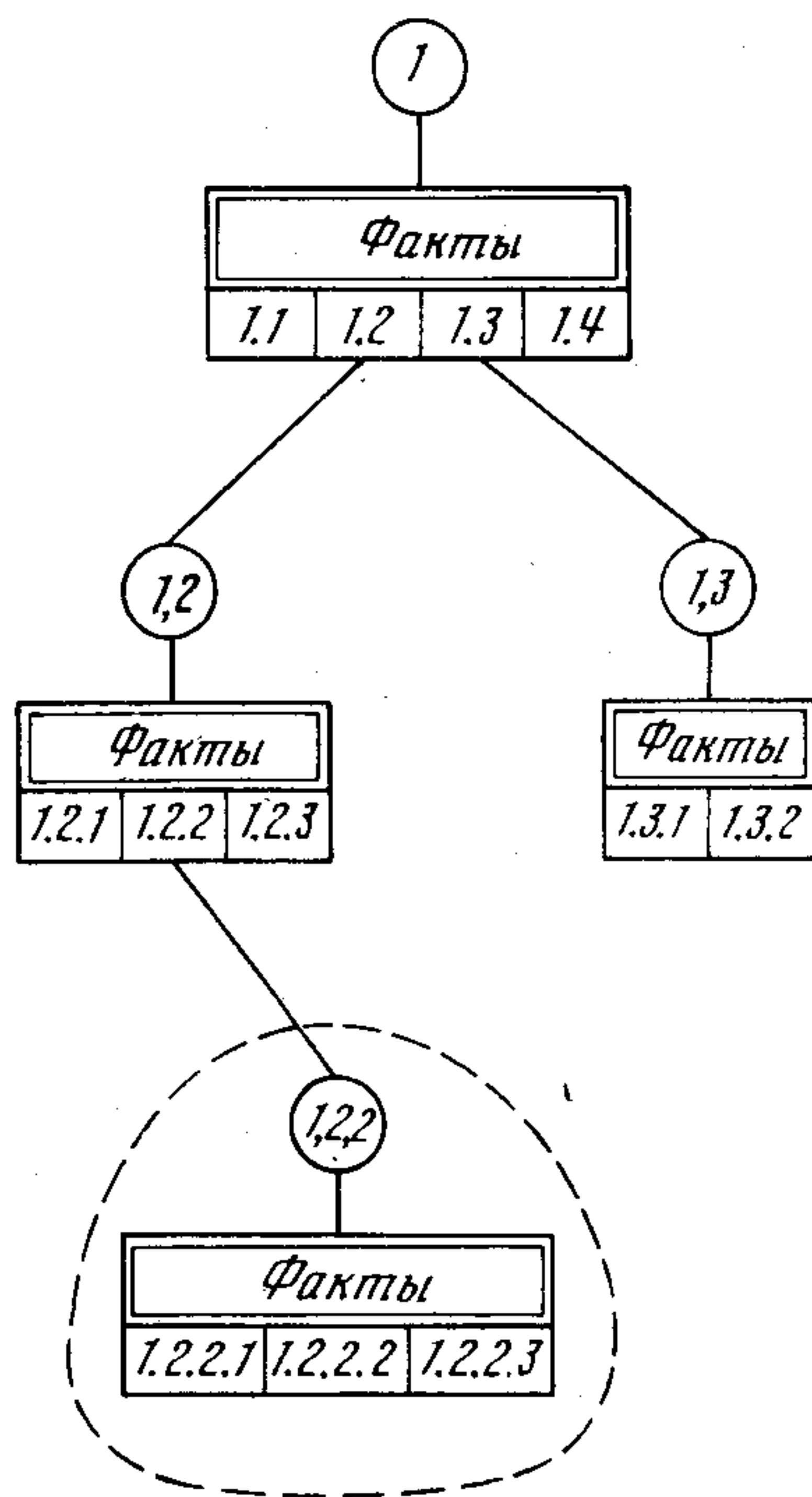


Рис. 1. Краевое сознание. Пунктиром обведено внимание системы

¹ При сопоставлении предлагаемых ниже механизмов с результатами самонаблюдения, по-видимому, разумно считать, что последовательность действий (как внешних, так и внутренних), совершаемых человеком сознательно, и обстоятельства, на которые он при этом обращает внимание, аналогичны последовательности действий активных блоков и потоку записей, проходящих через внимание системы.

Переключение внимания может происходить и в том случае, когда блок еще не решил задачу, не получил приказа об ее отмене и не поставил новую. Тогда внимание, содержащее нерешенную задачу, остается в составе дерева задач. (Именно поэтому внимания в дереве задач не упорядочены линейно.) Человек не осознает причин большинства переключений внимания такого рода. Бывает и так, что содержимое внимания долго кажется неизменным, а затем происходит внезапное переключение. Поэтому представляется вероятным, что автоматический (неосознаваемый) механизм, его осуществляющий, пользуется информацией, находящейся вне внимания.

В животном такое переключение основано на том, что изменения ценностей альтернативных действий-задач должны происходить не только внутри блока, активно работающего, решающего в данный момент задачу, но и в других блоках системы.

Если дерево задач содержит не менее двух узлов, то существуют блоки, на поле Z_0 внимания которых имеются надзадачи¹ задачи Z , находящейся сейчас во внимании системы. Такие блоки мы будем называть *ждущими*, а их действия, приведшие к тому, что сейчас решается именно задача Z , — *исполняемыми*.

На рис. 1 все блоки, кроме блока, решающего задачу 1.3, и активного блока, решающего задачу 1.2.2 (т. е. Z), — ждущие, причем, например, 1.2 — исполняемое действие в блоке 1.

В тот момент, когда действие A было выбрано для исполнения ныне ждущим блоком, оно имело, вероятно, максимальную ценность. Ясно, что, если затем A станет существенно уступать по своей ценности альтернативному действию B того же блока, то разумно прервать работу над Z и передать управление блоку, выбирающему между A , B и другими альтернативами.

Таким образом, внимание системы переходит к любому ждущему блоку, во внимании которого появилось действие, значительно более ценное, чем исполняемое им.

Этот механизм имеет много общего с методом упорядоченного поиска [6]. Можно надеяться, что выбираемая таким способом последовательность задач будет приводить в конечном счете к экономии времени.

2.2. *Изменение ценностей действий.* Как изменяются ценности действий в краевом сознании вне внимания системы? Рассмотрим сначала случай, когда показания рецепторов блока не меняются, и лишь растет время t , прошедшее с момента начала решения задачи. Тогда ценности всех действий, кроме исполняемого, не изменяются, а оценка исполняемого определяется формулой

$$W = W_0 f(t/T),$$

¹ *Надзадача задачи A* — это такая задача B , для которой в краевом сознании существует последовательность подзадач, приводящая к A .

где f — некоторая монотонно убывающая функция, определяющая степень «упрямства» системы. Здесь T — среднее время, которое раньше требовалось на выполнение действия, т. е. на решение подчиненным блоком соответствующей задачи, а величина W_0 прямо от времени не зависит и меняется лишь с изменением состояния внимания системы.

Зависимость такого рода отражает тот факт, что, если в течение какого-то времени в задаче «нет продвижения», т. е. во внимании системы не происходит существенных изменений, то ценность исполняемого действия во внимании блока-начальника автоматически падает.

Мы не будем приводить формулы, описывающие зависимость W_0 от состояния внимания системы, а ограничимся следующим замечанием. Ценность исполняемого действия будет увеличиваться в моменты постановки новых задач, а также при появлении во внимании системы фактов или любых записей (действий, описаний ситуации...), превосходящих по своей ценности его прежнее содержимое. Если же величина максимальной ценности падает, то падает и ценность исполняемых действий.

Уточнение целесообразных алгоритмов изменения ценностей станет, вероятно, возможным лишь после экспериментов с программой.

2.3. Потребности, внимание и ценности. При описании потребностей в п. 1.4 мы постулировали для каждой потребности существование своего счетчика, измеряющего «уровень потребности». Приписывание потребностям ценностей дает возможность вписать их в общую схему выбора очередной задачи. Мы предполагаем, что существует некий «верхний» блок, всегда находящийся в краевом сознании, выбирающий в качестве задачи системы удовлетворение одной из потребностей. Этот выбор производится в соответствии с ценностями потребностей, аналогично выбору действий в других блоках.

Во всех неактивных блоках, кроме верхнего, могут изменяться ценности только исполняемых действий. На ценности же потребностей, определяющие действия верхнего блока, это правило не распространяется. Например, ценность удовлетворения голода определяется в основном показанием рецептора «Голод» (см. п. 1.4), а не решаемой задачей.

Потребности являются корнями деревьев задач: когда ценность какой-нибудь из них становится достаточно большой, то она появляется во внимании системы, и ее удовлетворение становится текущей задачей, независимо от того, чем животное занималось перед этим.

Законы изменения ценностей потребностей определяют то, к чему животное стремится.

Над задачами, порожденными потребностями, работают, как правило, поочередно: взявшись за удовлетворение одной из потребностей, ищут действия, подходящие для этой узкой, временной

цели ¹. Правда, на этапе прогноза результатов намечаемого действия, кроме проверки близости прогноза к задаче, решаемой сейчас работающим блоком (п. 1.8), целесообразно проверять еще два обстоятельства. Во-первых, не будет ли мешать ожидаемый результат решению непосредственных предков этой задачи, во-вторых, не станет ли ситуация плохой, с точки зрения основных потребностей системы. И то и другое должно влиять на ценность действия.

Помимо потребностей типа голода, определенных в п. 1.4, ценности которых явно зависят от времени, можно ввести потребности, ценность которых становится ненулевой лишь при определенных показаниях внешних рецепторов. Мы допускаем также и «отрицательные» потребности, так что в нашей терминологии стремление избежать боли тоже является потребностью.

Напомним еще раз, что список потребностей не фиксирован. Его элементы могут появляться и исчезать. Каждая подзадача, не слишком легкая, но много раз бывавшая решенной для удовлетворения изначальной или ранее сформированной потребности, сама переводится в ранг потребности. С другой стороны, так возникшая потребность может и «угаснуть», исчезнуть из списка потребностей, например, если мир или животное изменятся так, что попытки удовлетворения этой потребности станут бесплодными или будут систематически приводить к «плохим» ситуациям с очень большими значениями ценностей других потребностей.

2.4. Влияние внешней ситуации на внимание. У человека внешняя ситуация может влиять на ценности фактов во внимании. У животного это получается автоматически, поскольку ценность факта (см. п. 1.8) зависит от текущей ситуации. Мы считаем, что в крайнем сознании имеется постоянно меняющееся изображение текущей ситуации — совокупность показаний внешних и внутренних рецепторов. Факты, лежащие во внимании активного блока, все время изменяют свои ценности, в зависимости от тех сторон ситуации, которые доступны рецепторам блока. Мы полагаем также, что изменения внешней ситуации должны влиять на ценности во внимании не только активного, но и всех ждущих блоков.

Во внимании системы всегда находится только часть описания ситуации, именно то, что воспринимается рецепторами активного блока. Из сказанного в п. 2.1 следует, что если изменение внешней ситуации вызвало переключение внимания, то во внимании системы появятся показания именно тех рецепторов, в

¹ Попытки удовлетворения потребностей порознь противоречат, конечно, тому, что животное хотело бы их все удовлетворить одновременно, и в принципе автомат, вычисляющий ценности подзадач какой-нибудь потребности, должен был бы учитывать и все остальные потребности, так, чтобы для исполнения выбирались действия, полезные для удовлетворения «в среднем» всей совокупности потребностей. Однако даже в столь сложных системах, как человек, это пожелание не удается реализовать полностью, и поведение определяется механизмом, называемым иногда принципом доминанты [7].

результате изменений выходов которых случилась приведшая к этому переключению «переоценка ценностей».

Разумеется, здесь речь идет лишь о произвольном обращении внимания на те или иные стороны ситуации. Произвольное же обращение можно рассматривать как одно из внутренних действий блока, которое должно выбираться по тем же законам, что и любое другое действие, причем его результат должен состоять в передаче управления, эквивалентной допуску во внимание системы всей или некоторой части текущей ситуации или, если это допускается конструкцией блока, в подключении ранее отключенных рецепторов.

3. Устройство памяти. Чтобы выбрать целесообразное действие, животное пытается вспомнить, что именно помогало «в аналогичных случаях». Если хранящиеся в памяти факты не упорядочены систематически и если каждый типичный факт — руководство к будущим действиям — записан лишь однократно, то для нахождения нужного факта одному последовательно работающему процессору пришлось бы просматривать каждый раз в среднем половину всех хранящихся фактов, что, конечно, нежелательно.

Сегодня нам известны два пути, позволяющие резко сократить время поиска нужных фактов. Первый — это создание на множестве фактов в памяти некоторой структуры, которая направляла бы поиск так, чтобы каждый раз просматривалась лишь малая часть памяти. Второй — это организация параллельного поиска в памяти. Вероятно, избрание одного из этих путей уже в очень значительной степени определяет всю модель. Мы пошли по первому, лучше изученному пути, хотя не исключено, что именно второй приведет к более красивой и единообразной системе.

Животное включает в себя ряд блоков, имеющих отдельные памяти. Опишем устройство памяти блока: сначала — форму отдельных записей-фактов, затем — их организацию в структуру.

3.1. Форма необработанного факта. Рассмотрим вначале факт, являющийся непосредственной и необработанной записью происшедших с блоком событий. Такой факт состоит из четырех частей:

1) описания ситуации, в которой находился блок, т. е. утверждения вида $S_0 = A$,

2) описания стоявшей перед ним задачи — утверждения вида $Z_0 = B$,

3) описания выполненного действия — утверждения вида $D_0 = C$,

4) описания ситуации K , в которой очутился блок спустя некоторое время после совершения действия.

Здесь S_0 , Z_0 и D_0 — названия выделенных полей внутри внимания, на которых находятся текущая ситуация, задача и действие, выбранное блоком. Таким образом, факт будет иметь вид:

было $(S_0 = A, Z_0 = B, D_0 = C)$, получилось ¹ K ,

¹ Может показаться, что данное определение необработанного факта противоречит нашему внутреннему опыту: воспоминания не имеют такого вида. Однако ниже будет показано, что в модели необработанные факты редко

или, если ввести специальные метки S_1 и S_2 ,
 $S_1 : (S_0 = A, Z_0 = B, D_0 = C); S_2 : K.$

Заметим, что при проверке применимости такого факта, извлеченного из памяти, на метку S_1 не обращают внимания и проверяют только утверждения в скобках. Это является формальным выражением априорного правила перенесения прошлого опыта на будущее, без которого вообще невозможно использование памяти. Вместе с тем, при обработке фактов во внимании метка S_1 важна: она позволяет отличать утверждения о текущем состоянии блока от воспоминаний. Потеря этой метки может привести к галлюцинаторному восприятию и поведению.

3.2. Общий вид факта. В памяти блока будут также находиться факты иного происхождения и вида. Во-первых, это «врожденные знания» — информация, вложенная до начала «жизни» модели, при ее конструировании. Во-вторых, это обобщенные факты, построенные самой системой, которая обрабатывает совокупности записей о происходящем, выделяя существенные признаки, строя функциональные зависимости и т. д.

В п. 1.8 мы говорили, что память должна использоваться при поиске действия и прогнозе. Чтобы обработанный факт мог быть использован при поиске действия, он должен содержать условия, определяющие, применимо ли рекомендуемое им действие в данной ситуации для решения данной задачи. (В необработанном факте такими условиями являются равенства $S_0 = A$ и $Z_0 = B$.) Очевидно, эти условия есть утверждения о содержимом полей S_0 и Z_0 . Их мы будем обозначать $f_1(S_0, Z_0)$, где f_1 — некоторый предикат. Кроме этих условий в факте должна быть часть, дающая информацию о самом действии, выбираемом в зависимости от текущей ситуации и задачи. Эта часть должна иметь вид $D_0 = \varphi(S_0, Z_0)$, где φ — функция от описаний S_0 и Z_0 , значение которой — описание действия.

Можно ли ограничиться этими частями факта? Очевидно, нет, так как память должна применяться и для прогноза, в ходе которого механизм памяти должен пользоваться описанием текущей ситуации S_0 и описанием планируемого действия D_0 , а не описанием задачи Z_0 . Поэтому в факте, используемом для прогноза, должно быть условие применимости типа $f_2(S_0, D_0)$. С другой стороны, результатом прогноза должно быть описание получающейся после действия ситуации, зависящее от S_0 и D_0 . Поэтому прогноз, который мы получаем из факта, должен задаваться выражением типа $S_2 = \psi(S_0, D_0)$. Таким образом, факт, применимый и для прогноза и для выбора действия, должен состоять из четырех частей — $S_1 : (f_1(S_0, Z_0), D_0 = \varphi(S_0, Z_0), f_2(S_0, D_0)), S_2 : \psi(S_0, D_0)$.

используются при решении каких-либо задач. Можно думать, что обращение внимания на такие факты, их вспоминание, соответствует относительно редким у большинства взрослых людей случаям, про которые говорится: «вспомнил — как будто наяву увидел». В некоторых из таких ярких воспоминаний воспроизводится не только окружающая обстановка, но и наши стремления и наши действия с их результатами.

3.3. *Анализ факта при выборе действия.* Как нам кажется, при ограниченных возможностях блока обобщения нередко будет возникать положение, когда не удастся получить соотношение между S_0 и Z_0 , которое являлось бы надежным достаточным условием достижимости Z_0 с помощью определенного действия $D_0 = \varphi(S_0, Z_0)$, но удастся выразить результат S_2 через ситуацию S_0 и действие D_0 и отдельно выразить рекомендуемое действие как функцию S_0 и Z_0 . В этом случае описанное устройство факта дает возможность использовать предикат $f_1(S_0, Z_0)$ в качестве необходимого, но не достаточного условия применимости факта. Если это условие выполняется, то следует вычислить $D_0 = \varphi(S_0, Z_0)$, затем, если предикат f_2 этим D_0 удовлетворен, вычислить $S_2 = \psi(S_0, D_0)$ и, наконец, проверить, является ли S_2 решением задачи Z_0 . Очевидно, этот процесс определяет условие применимости факта в два этапа. При этом D_0 является промежуточной переменной, введение которой облегчает построение предикатов такой сложности.

Описанный процесс позволяет использовать факт и в тех случаях, когда предикат $f_1(S_0, Z_0)$ не выполняется, но «малое» изменение S_0 или Z_0 может привести к выполнению предиката. Для этого введем числовую меру $\rho(f)$ выполнения условий f , равную нулю, когда все условия в f выполняются точно, и тем большую, чем в большей степени условия не удовлетворяются [1]. Если для какого-то факта значение $\rho(f_1)$ меньше некоторого порога, то будем вычислять S_2 с помощью описанного процесса. Далее будем вычислять $\bar{\rho}$, показывающее, насколько это S_2 соответствует текущей задаче Z_0 . Ценность этого факта вычисляется как функция от $\bar{\rho}$, $\rho(f_1)$ и статистики применений факта (п. 1.8).

Таким образом, мы будем использовать факты в более широком классе ситуаций, чем тот, в котором они наблюдались. Очевидно, конструкция меры ρ определяет не меняющийся с развитием системы способ обобщения, эквивалентный заданию жесткой «таблицы различий» и их весов, в смысле [8].

3.4. *Структура памяти и поиск.* Наличие в фактах условий применимости позволяет придать памяти структуру, сокращающую поиск. На вершине этой структуры находятся факты с максимальной областью применимости, чем ниже — тем более узко применимые. Будем говорить, что факт A является *потомком* факта B , если из применимости A следует применимость B . В структуре памяти факт A будет связан только с его непосредственными потомками, т. е. с такими потомками B_i , что в памяти не существует факта C , являющегося предком B_i и потомком A . Для данных S_0 и Z_0 поиск в памяти при выборе действия начинается с факта на вершине структуры. Затем проверяются его непосредственные потомки. Ветви, идущие от тех фактов, которые неприменимы к текущим S_0 и Z_0 , далее не рассматриваются, так как ведут к заведомо неприменимым фактам. Если найден один из применимых потомков, то с ним повторяется та же процедура. Если же потомков нет, то возвращаемся вверх на один шаг.

Таким образом, производится поиск «сначала вглубь» [6] по имеющейся структуре. Поиск заканчивается, если исчерпана память, либо если найден факт, обещающий решить задачу. Поскольку в каждом узле структуры может происходить обрыв ветвей, при хорошей ее организации перебор будет невелик. Если система уже достаточно обучена и накоплено много эффективных и широко применимых фактов, то поиск редко будет доходить до нижних уровней структуры, и во внимании редко будут появляться необработанные факты, содержащие подробные описания событий прошлого.

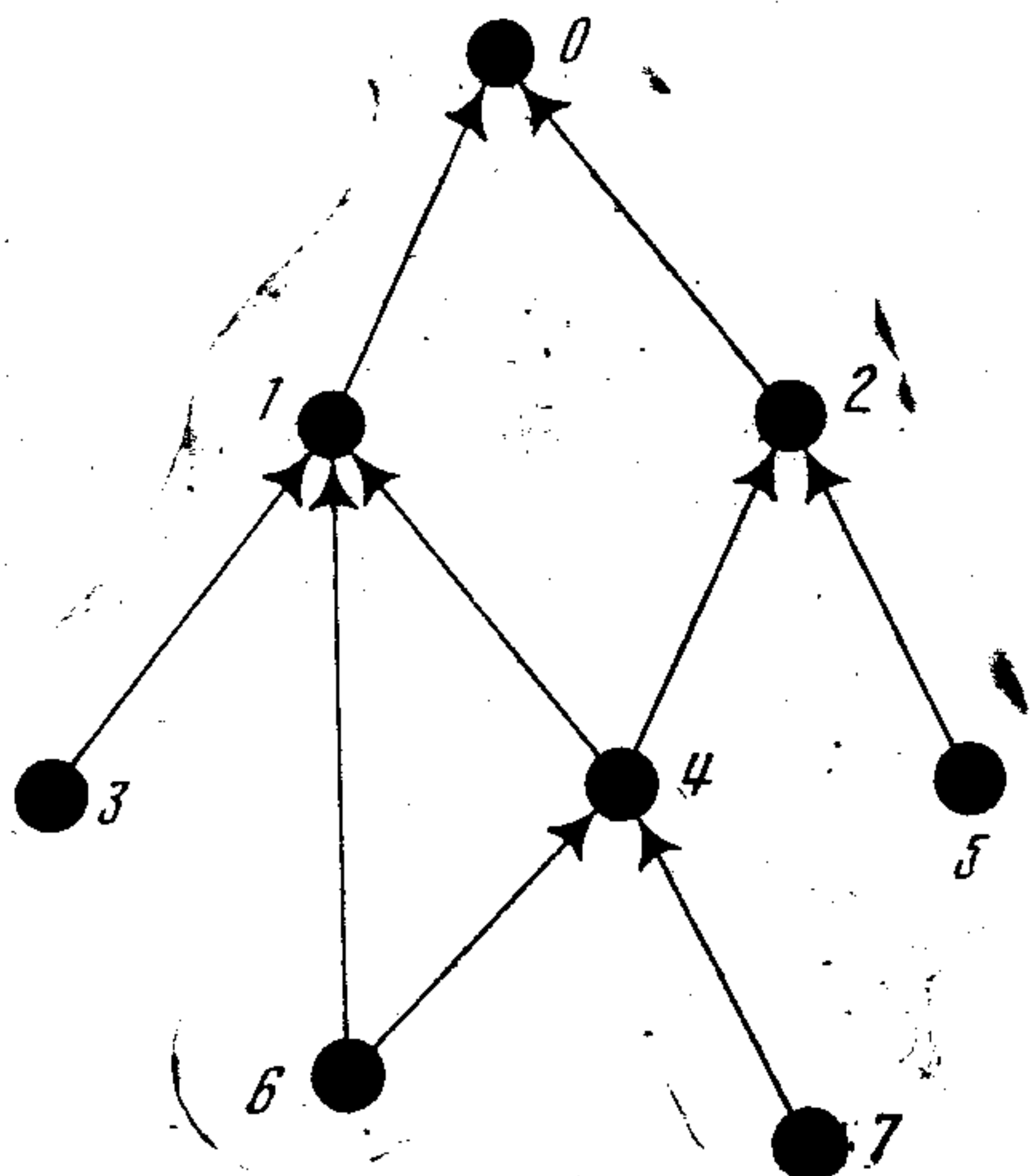


Рис. 2. Пример структуры памяти. Вершины графа соответствуют отдельным фактам в памяти

ряться суммой тех частей времени активной работы блока, когда ценность факта была выше пороговой. Каждый новый факт должен попадать в определенное место структуры, находящееся между его непосредственными предками и потомками.

Для этого с новым фактом проходят от вершины структуры вниз, выбирая в каждом узле пути, ведущие к фактам, более широко применимым, чем он. Те узлы, откуда нет пути вниз, и будут его непосредственными предками.

Затем следует установить связи факта с его непосредственными потомками, что обычно приводит к необходимости движения «снизу вверх».

3.6. Каталог. Поиск в памяти состоит из операций двух типов: 1) проверки применимости данного факта и 2) нахождения в памяти его непосредственных потомков (иногда и предков). Вторая операция должна производиться с помощью некоторого каталога, хранящего информацию о связях в структуре. На рис. 2 изображен простой пример структуры (стрелки проведены от каждого факта к его непосредственному предку), а в таблице — предлагаемый нами каталог, задающий эту структуру.

3.5. Запоминание фактов. Запоминать имеет смысл только неожиданные либо важные факты. Если результат действия по существенным параметрам отличается от прогноза, то заготовка факта, состоящая из описания начальной ситуации, задачи и действия (с добавлением его результата), объявляется неожиданной и заносится в память.

Все заготовки, которые превратились в факты, уже не сулящие неожиданностей, через некоторое время либо уничтожаются, либо, если факт оказался важным, переносятся в долговременную память. Важность факта для блока может изме-

Адрес	Факт	Сын	Отец	Брат	Двой- ник	Адрес	Факт	Сын	Отец	Брат	Двой- ник
a_0	A_0	a_1	—	—	—	a_5	A_4	a_7	a_1	—	a_6
a_1	A_1	a_2	a_0	a_3	a_1	a_6	A_4	a_7	a_3	a_8	a_5
a_2	A_3	—	a_1	a_4	a_2	a_7	A_6	—	a_6	a_9	a_4
a_3	A_2	a_5	a_0	—	a_3	a_8	A_5	—	a_3	—	a_8
a_4	A_6	—	a_1	a_6	a_7	a_9	A_7	—	a_6	—	a_9

В первом столбце таблицы помещены машинные адреса строк каталога. Каждой строке каталога соответствует факт, адрес которого в основной памяти указан во втором столбце. Чтобы от факта можно было двигаться к его потомкам, в столбце с названием **Сын** указан каталожный адрес строки, соответствующей первому непосредственному потомку факта. Чтобы от факта можно было двигаться к его предкам, в столбце, названном **Отец**, имеется каталожный адрес строки, соответствующей одному из непосредственных предков факта.

Для того чтобы найти всех непосредственных потомков факта A , нужно в соответствующей строке взять адрес строки **Сын** a , где в столбце **Факт** имеется адрес первого непосредственного потомка A , а в столбце **Брат** — каталожный адрес строки, соответствующей следующему непосредственному потомку того же A . В этой новой строке в столбце **Брат** отыщем адрес строки для следующего потомка A и т. д.

Упомянутых столбцов было бы достаточно, если бы изображаемая структура была деревом. Однако это не так, и у факта может быть несколько непосредственных предков. Поэтому в каталог вводится столбец **Двойник**. В нем для строки a_i , соответствующей факту с адресом в основной памяти A_k , имеется адрес a_j , снова отсылающий к A_k . Однако в строке a_j факт A_k рассматривается уже как член другого семейства, т. е. в столбце **Отец** имеется ссылка на каталожный адрес другого непосредственного предка, и, соответственно, ссылка в столбце **Брат** тоже другая. Пользуясь столбцами **Отец** и **Двойник**, можно найти всех непосредственных предков данного факта. Легко показать, что каталог описанного типа адекватен структуре памяти.

Нам кажется, что движение по такому каталогу будет отнимать намного меньше времени, чем проверка применимости факта. Поэтому установление неприменимости какого-нибудь факта можно использовать не только для отказа от просмотра его потомков, но и для пометки их в каталоге метками неприменимости. Это имеет смысл, если структура памяти не является деревом, так как позволяет не тратить времени на проверку применимости помеченных потомков, к которым приходят по другому пути. Аналогично, установив применимость некоторого факта, мы можем отметить метками применимости всех его предков.

Если структура памяти далека от дерева (система имеет богатый набор «ассоциаций»), то поиск будет быстрым, так как потребуется мало проверок на применимость. Это соответствует нашим представлениям о том, что богатство ассоциаций — отнюдь не порок, но важное преимущество структуры памяти.

3.7. Условия образования эффективной структуры памяти. Для быстрого поиска каждый факт должен иметь в среднем по два

потомка, и, если фактов много, то глубина структуры велика. Чем выше факт расположен в структуре, тем шире область его применимости. Таким образом, для хорошей организации памяти необходимы эффективные алгоритмы, строящие факты с широкими областями применимости.

Однако этого мало. Дело в том, что в зависимости от задачи меняется набор существенных для решения рецепторов. Вариантов таких наборов может быть очень много. Факты, в условиях применимости которых упоминаются различные рецепторы, не могут быть потомками друг друга. Поэтому на любом уровне обобщения структура будет сильно разветвленной, так что перебор всех ветвей в произвольном порядке сильно замедлит поиск. Желательно, чтобы в узлах с большим числом ветвей выбор с большой вероятностью падал на нужную ветвь. Блочная структура дает такую возможность. Действительно, память каждого блока будет содержать показания относительно небольшого набора рецепторов, и, после обучения, достаточно однородный набор задач, успешно решаемых данным блоком. Поэтому степень ветвления в структуре памяти блока будет невелика. Сам переход от блока к блоку тоже можно рассматривать как выбор части памяти для дальнейшего поиска. Если количество подчиненных, которым блок может передать управление, велико, то, с точки зрения структуры памяти, это соответствует узлу с многими ветвями. Однако в этих случаях выбор ветви, т. е. передача управления, входит в действие блока-начальника, которое производится не случайно, а на основании прошлого опыта.

Мы думаем, что модель, предназначенную для решения разнородных задач, легче построить, если разбить задачи на группы, требующие определенных знаний и умений, а овладение каждой группой поручить специальному блоку, снабженному соответствующим набором рецепторов и доступных действий. Однако достичь хорошего соответствия возможностей блока порученным задачам без опыта трудно. Поэтому должны быть предусмотрены средства взаимодействия блоков и частичного перемещения фактов из памяти одного блока в другой.

3.8. Обмен информацией между блоками. Блок-начальник должен уметь приказывать — формулировать задачу для подчиненного. Но разные блоки обладают разными наборами рецепторов и воспринимают разные стороны ситуации, в которой находится животное. Поэтому возникает проблема перевода задачи, стоящей перед начальником, на языки подчиненных: неопытный начальник не знает, к какой ситуации должен стремиться подчиненный, и формулирует задачу неверно. В этом случае задача, стоящая перед начальником (Z_n), может быть решена лишь случайно. Если при этом окажется решенной и задача подчиненного (Z_p), то для решения Z_n начальнику следует и впредь приказывать добиваться Z_p .

Однако случайное действие D_p , давшее Z_n , может не решить Z_p . Для закрепления удач такого рода полезно в память началь-

ника записать факт, действием которого служит постановка в качестве задачи ситуации S^* , в которой оказался подчиненный после действия $D_{\text{п}}$.

С другой стороны, в памяти подчиненного, решившего своим действием задачу начальника, можно записать факт, в котором, кроме его собственной S^* , будет указана задача, поставленная его начальнику. После этого подчиненный сможет выбирать правильное действие, даже если начальник не будет заботиться о переводе своей задачи на другой язык.

4. Построение цепочек действий. В животном предусмотрен механизм планирования цепочек только из двух действий¹. Он по текущим $S_0 = A$ и $Z_0 = B$ ищет в памяти блока пару фактов Φ_1 и Φ_2 таких, что Φ_1 применим к A , а Φ_2 применим к ситуации, возникающей в результате выполнения действия из Φ_1 , и, после совершения действия из Φ_2 , Z_0 выполняется. Пару Φ_1 и Φ_2 мы будем называть *цепочкой фактов*, а задачу, решаемую действием из Φ_1 , — промежуточной задачей Z^* .

Найдя такую цепочку, блок ставит самому себе задачу Z^* , как Заказчик Исполнителю, так что, решив Z^* , он автоматически возвращается к Z_0 .

По нашему мнению, достаточно иметь механизм, строящий цепочки только из двух фактов, если запоминать *цепочечный факт* вида $S_1 : (S_0 = A, Z_0 = B, D_0 = \text{себе } Z^*), S_2 : (S^*, \text{ затем } B)$.

Таковыми цепочечными фактами можно пользоваться как и любыми другими, в частности, для построения более длинных цепочек действий и соответствующих им фактов.

Когда в памяти системы накопится много цепочечных фактов, то большая их часть будет соответствовать часто встречающимся задачам. Если эти факты удастся обобщить и они станут широко применимыми, то из них при двухшаговом планировании можно будет строить планы, содержащие много элементов, с помощью перебора, небольшого в сравнении с тем, который потребовался бы при построении таких планов непосредственно из элементарных действий.

При такой схеме «короткого планирования», запоминания и обобщения модель, вероятно, будет сходна с человеком и животными в том, что сможет решать сложные задачи лишь после тренировки — решения умело подобранной последовательности постепенно усложняющихся задач. Заметим, что человек способен планировать и длинные цепочки действий, но только в знакомых областях, когда для каждой поставленной задачи есть подзадача, про которую «чувствуется», что ее стоит решать. (В нашей схеме это означает, что в памяти имеется цепочечный факт с большой ценностью, рекомендующий поставить определенную подзадачу.) Человек же, работающий в непривычной области, редко создает в уме многозвенные планы, а после продумывания двух-трех шагов переходит к пробам во вне.

¹ Для планирования длинных цепочек действий созданы сложные системы, такие как STRIPS [9] и LAWALY [10]. Поскольку блок, решая задачу Z^* , снова может строить цепочку, принцип двухшагового планирования в действительности не ограничивает глубину плана.

Описанная выше схема использования цепочечного факта требует многократной смены задач во внимании и поиска в памяти, что отнимает время. Этого можно избежать, введя механизм, который после многократного успешного использования цепочки с промежуточной задачей Z^* строит новый факт Φ_0 . Область применимости Φ_0 определяется множеством ситуаций и задач, в которых цепочка $\Phi_1\Phi_2$ оказывалась полезной, а действие Φ_0 заключается в передаче управления некоторой процедуре. Работа этой процедуры описывается так:

- 1) соверши то, что делал в Φ_1 ,
- 2) если в положенный срок получилось Z^* , то перейди к 3), если не получилось, — то к 4),
- 3) соверши то, что делал в Φ_2 , сообщи Заказчику «исполнено» и отдай ему управление,
- 4) помести во внимание Заказчика сообщение о неудаче первого шага и отдай ему управление.

Такие процедуры мы будем называть «автоматизмами». Они занимают промежуточное положение между «слепыми» эффекторами и обучаемыми блоками, и, как нам кажется, похожи по проявлениям в поведении на автоматизмы, описанные Н. А. Бернштейном [11].

5. Прогноз и модель мира. Рассмотрим теперь случай, когда для намеченного действия не удастся найти прогноз. Причина может состоять в том, что в памяти нет фактов, подходящих к текущей ситуации. Тогда, как это описано в [2], полезно обратиться к обобщению. Однако возможен случай, когда самое хорошее обобщение необработанных фактов ничего не дает, поскольку в необработанный факт в качестве начальной ситуации заносятся показания органов чувств лишь на момент начала действия, в то время как результат может зависеть от далекой предыстории. Поэтому разумно предположить, что знание одной текущей ситуации дает возможность получить лишь *частичный прогноз* \tilde{S} конечной ситуации.

Простейший пример: если радиус видимой части мира конечен и требуется предсказать, что мы увидим, сдвинувшись на шаг, то необходимо знать, какие предметы находятся вне нашего поля зрения. В работе [12] кроме «аксиомной модели мира», описывающей результаты действий робота, введена еще «решетчатая модель среды» [12, 13], согласно которой для каждой точки среды, где бывал робот, записывается определенный набор сведений. При прогнозе робот использует эти описания.

Очевидно, для эффективного прогноза необходимо иметь в памяти факты, в которых функции, определяющие конечную ситуацию, зависят не только от текущей ситуации и намеченного действия, но и от описаний давно прошедших событий.

В животном такие описания предполагается получать из описаний конечных ситуаций в старых фактах. Среди них можно найти такое описание S , частью которого является частичный прогноз

S , и, в простейшем случае, взять это S в качестве прогноза. В менее примитивной модели прогноз следует искать как функцию от S и тех ситуаций прошлого, в которые входила S . Эта функция должна быть найдена самой моделью в процессе обучения.

Заключение. На этом мы заканчиваем описание общей схемы животного. Остались нерассмотренными не только многие трудные технические вопросы, но и ряд принципиальных задач: алгоритмы обобщения, процедуры работы над фактами внутри внимания, правила приписывания ценностей и т. д. Однако нам кажется, что и на этом этапе уже бесполезно сравнивать свойства получающейся модели с известными особенностями поведения человека.

Так, нам представляется интересным, что некоторые черты человеческой психики оказываются в модели следствиями стремления к оптимальной конструкции. Например, нецелесообразные, на первый взгляд, произвольные переключения внимания — следствие перевычисления ценностей в краевом сознании, которое является частью процедуры оптимизации распределения усилий системы.

Мы надеемся, что, если уже такое умозрительное рассмотрение свойств модели дает возможность высказать показавшиеся нам интересными предположения о строении психики человека, то завершение модели и опыты с нею на ЭВМ послужат поводом для появления теорий поведения, связывающих воедино гипотезы об отдельных психических функциях.

Л и т е р а т у р а

1. М. М. Бонгард, И. С. Лосев, В. В. Максимов, М. С. Смирнов. Формальный язык описания ситуаций, использующий понятие связи.— Наст. сб.
2. И. С. Лосев, В. В. Максимов. О задаче обобщения начальных ситуаций. Наст. сб.
3. М. М. Бонгард. Проблема узнавания. М., «Наука», 1967.
4. Ж. Адамар. Исследование психологии процесса изобретения в области математики. М., «Советское радио», 1970.
5. У. Джемс. Психология. СПб., 1911.
6. Н. Нильсон. Искусственный интеллект. М., «Мир», 1973.
7. А. А. Ухтомский. Письма.— Сб. «Пути в незнание». М., «Советский писатель», 1973.
8. А. Ньюэлл, Дж. Шоу, Г. Саймон. Разновидности интеллектуального обучения «Вычислителя для решения задач общего типа».— Сб. «Самоорганизующиеся системы». М., «Мир», 1964.
9. Р. Файкс, Н. Нильсон. Система STRIPS — новый подход к применению методов доказательства теорем при решении задач. Сб. «Интегральные роботы». М., «Мир», 1973.
10. L. Siklossy, J. Dreussi. An Efficient Robot Planar which generates its own procedures. In: Third International Joint Conference on Artificial Intelligence. Palo-Alto (Cal.), 1973.
11. Н. А. Бернштейн. О построении движений. М., «Медгиз», 1947.
12. Н. Нильсон. Мобильный автомат, построенный с использованием принципов искусственного интеллекта.— Сб. «Интегральные роботы». М., «Мир», 1973.
13. Дж. Доран. Планирование и роботы.— Сб. «Интегральные роботы». «Мир», 1973.