

ВВЕДЕНИЕ

Человек всегда старался понять, как он мыслит, по каким законам и правилам строятся его рассуждения, в чем гарантия истинности его умозаключений. Логика - древнейшая из наук. Законы, открытые еще Аристотелем (IV век до н.э.), верны и поныне и лежат в основе современной формальной логики. В XX веке логика как наука приобретает второе дыхание - она становится наукой... технической: по законам формальной логики строятся электрические схемы элементов и узлов ЭВМ, позволяя им решать не только математические, но и чисто логические задачи. Все это в какой-то мере моделировало интеллектуальную деятельность человека. Но хотелось большего.

Проследим вкратце эволюцию развития «интеллекта» ЭВМ (машинного интеллекта). В ней можно выделить несколько принципиально различных этапов развития.

Этап 1 (50-е – 60-е годы). Он связан с появлением первых машин последовательного действия, с очень небольшими, по сегодняшним меркам, ресурсными возможностями по памяти, быстродействию и классам решаемых задач. Это были задачи сугубо вычислительного характера, для которых были известны схемы решений и которые можно описать на некотором формальном языке. К этому же классу относятся и задачи на адаптацию.

Таким образом, первый этап эволюции «интеллекта» ЭВМ соответствовал формированию вычислительного «интеллекта», который носил чисто алгоритмический характер и достаточно просто поддавался формализации и программированию. Поэтому, при разработке систем автоматизации различных видов деятельности человека, необходимо было: выделить задачи расчетного характера, увязать их в определенную систему, построить формальные математические модели, подобрать методы решения, запрограммировать, проверить на адекватность и соответствие результатам деятельности человека и затем использовать в практической работе.

Можно сказать, что такого рода автоматизация носила фрагментарный (точечный) характер и, естественно, быстро перестала удовлетворять реальным потребностям. Недостатком такого рода алгоритмов было то, что в них в едином алгоритмическом процессе переплетались данные входные и промежуточные с операторами их

обработки (действиями). И отделить данные от процессов их преобразования (знаний о процессах обработки) было невозможно. Это создавало огромные потребности в программистах и необходимость постоянного «дотягивания» программ до постоянно меняющихся алгоритмов. И даже появление алгоритмических языков, которые повышали производительность труда программистов не менее, чем на порядок, ситуация принципиально не изменилась.

Этап 2 (60-е – 70-е годы). По мере развития возможностей ЭВМ и их насыщения, усложнялись задачи и осознавались новые рутинные виды деятельности, которые хотелось передать машине. В частности, был осознан новый класс задач, решаемых человеком постоянно. Это были задачи поиска и сортировки информации, работа с БД. В «интеллект» машины добавились механизмы поиска, сортировки, простейшие операции по обобщению информации, не зависящие от смысла обрабатываемых данных. Это стало новой точкой отсчета в развитии и понимания задач автоматизации деятельности человека. Можно сказать, что с этого момента и началась эпоха массовой автоматизации систем управления, проектирования, управления технологическими процессами и т.п., продолжавшаяся до конца 80-х годов.

Этап 3 (70-е – 80-е годы). Практически параллельно в мире (США, Англия) развивалось научное направление, идеи которого принципиально отличались от принятых. Задачи этого класса характеризуются отсутствием априори известных схем решения, даже при условии привлечения дополнительных знаний о предметной области. Поэтому для решения таких задач используются сложные иерархические построения и программы, имитирующие механизмы мышления человека. Этот класс задач было принято считать интеллектуальным, а методы их решения стали развиваться в рамках научного направления, получившего название эвристического программирования. Его сторонники (Саймон А., Шоу Дж., Геленгер Г. и др.) считали, что интеллектуальной можно считать только ту программу, которая обладает механизмом логического вывода и которая на его основе способна самостоятельно синтезировать алгоритмы решения задач. При этом под решением задач понимался такой алгоритм, выполнение которого переводит объект управления (процесс, среду и т.п.) из начального состояния в целевое.

Этот алгоритм необходимо было каждый раз синтезировать автоматически на основе описания пространства состояний задачи.

При этом в качестве прототипа модели механизма мышления использовалась теория логического вывода, хорошо разработанная в метаматематике.

На этом пути были достигнуты фундаментальные для теории ИИ результаты: апробирован и исследован механизм логического вывода и разработаны программы автоматического доказательства теорем в логике, арифметике, планиметрии; получены новые методы и алгоритмы, ускоряющие процесс логического вывода и сделавшие возможным его эффективную реализацию на ЭВМ и др. Именно в рамках эвристического программирования были разработаны: теория построения пространства состояний; методы поиска решений в нем (лабиринтная гипотеза); наработаны универсальные эвристики разбиения задачи на подзадачи и сокращения пространства поиска решений.

Это стало основой новых методов, языков программирования и моделей постановки и решения оптимизационных задач в математическом программировании (методы скорейшего спуска, ветвей и границ и др.), в распознавании образов.

Другим важнейшим результатом этого этапа развития ИИ стало понимание возможности разделения данных и механизмов их обработки в рамках формальных моделей представления знаний. Развитие этого направления стимулировалось наличием социального заказа в США, Японии и Европе на создание интеллектуальных роботов для исследования космического пространства и новых планет, для работы в недоступных или опасных для человека средах. Построить искусственный разум робота для таких целей невозможно было на основе вычислительного и поискового интеллектов. Нужен был принципиально новый подход, который развивался в рамках эвристического программирования и моделирования нейронных сетей (персептронов). Исследования проводились на задачах: на кубиках (планирование поведения роботов при сборке простейших конструкций), на построение пространства состояний и поиск в нем решений для игр (крестики – нолики, шашки, шахматы и т.п.)

В рамках этих исследований появились понятия: модель предметной области, пространство состояний и поиска решений, модель представления знаний, стратегии вывода и др., которые и составили в дальнейшем предмет нового научного направления под общим названием «искусственный интеллект». Таким образом, третий этап развития обогатил «интеллект» машины логикой

(логическим выводом), как одним из основных механизмов мышления человека.

Этап 4 (80-е – 90-е годы). В конце 70-х годов в мире даже наметился некоторый "кризис жанра", поскольку огромные вложения не приводили к сколько-нибудь убедительным практическим результатам. Это продолжалось до тех пор, пока учеными не была осознана важность *знаний* (по объему и содержанию) для синтеза интересных алгоритмов решения задач. При этом имелись в виду знания, с которыми математика не умела работать, т.е. опытные знания, не носящие строгого формального характера и описываемые обычно в декларативной форме. Это знания специалистов в различных областях деятельности, врачей, химиков, исследователей и т.п. Такие знания получили название экспертных знаний, и соответственно системы, работающие на основе экспертных знаний, стали называться системами-консультантами или экспертными системами (не путать с экспертными оценками - это совсем другое).

В рамках экспертных систем, как важнейшего типа систем ИИ, произошло соединение неформально описываемых знаний с механизмами логического вывода и другими методами, разработанными в рамках эвристического программирования. Это дало потрясающий эффект в виде экспертных систем для самых различных направлений деятельности, к которым ранее никакие математические формализации не подходили, как-то: медицина, химия, геология, управление и т.д. Именно в рамках экспертных систем были разработаны неформальные модели представления знаний, модели вывода в различных представлениях, псевдофизические логики, логика рассуждений или логика правдоподобного вывода и т.п.

Четвертый этап развития ИИ стал прорывным. С появлением экспертных систем в мире начался принципиально новый этап развития интеллектуальных технологий – эра интеллектуальных систем - консультантов, которые предлагали варианты решений, обосновывали их, способны были к обучению и, следовательно, к развитию, общались с человеком на привычном для него, хотя и ограниченном, естественном языке. Знания стали товаром. Носителей систем знаний называли экспертами. Человечество получило возможность сохранять и накапливать базы знаний отдельных специалистов (или групп специалистов) в определенной области. Знания стало возможным собирать, тиражировать, проектировать,

сделать доступным для всех заинтересованных в нем людей. Появилась новая профессия – «инженер по знаниям» или «инженер - когнитолог».

Этап 5 (1990 – 2000 годы). Усложнение систем связи и решаемых задач потребовало качественно нового уровня «интеллектуальности» обеспечивающих программных систем, таких систем, как защита от несанкционированного доступа, информационная безопасность ресурсов, защита от нападений, смысловой анализ и поиск информации в сетях и т.п. И новой парадигмой создания перспективных систем защиты всех видов, стали интеллектуальные системы. Именно они позволяют создавать гибкие среды, в рамках которых обеспечивается решение всех необходимых задач. Это новое направление получило название мультиагентных систем. Каждый агент имеет свою систему целеполагания и мотивации, свою область действий и ответственности, а взаимодействие между ними обеспечивается метаинтеллектом. В рамках такого осмысления традиционные методы, алгоритмы и программы становятся элементарными «кирпичиками», из которых строятся затем алгоритмы и решения возникающих задач. Таким образом моделируется некоторое виртуальное сообщество интеллектуальных агентов-систем, которые автономны, активны, вступают в различные «социальные» отношения – кооперации и сотрудничества (дружбы), конкуренции, соревнования, вражды и т.п. Этот «социальный» аспект решения задач - *фундаментальная особенность концептуальной новизны передовых информационных технологий, искусственных (виртуальных) организаций, виртуального общества.*

Уже сегодня мультиагентные системы находят широчайшее применение для: распределенного решения сложных задач, совмещенного проектирования изделий, построения виртуальных предприятий, моделирования больших производственных систем и электронной торговли, электронной разработки сложных компьютерных систем, управления системами знаний и информации и т.п. Мы являемся свидетелями триумфального шествия по миру интеллектуальных информационных технологий (ИИТ), основанных на идеях экспертных систем и мультиагентности. Именно они делают цивилизованные страны практически недостижимыми для других, пока еще только развивающихся, стран. Именно ИИТ сегодня в мире

являются источником глобальной власти и материального благополучия. Борьба пере

В данной книге ставится цель краткого, но систематического изложения фундаментальных основ теории ИИ, без которых невозможно дальнейшее продвижение к пониманию современных прикладных интеллектуальных систем.

Так, в первом разделе даются основные понятия ИИ, такие как: предметная область, модель предметной области, состояние (начальное, целевое и др.), решение, пространство состояний. Далее, во втором разделе, рассматриваются основные методы поиска решений в пространстве состояний.

В третьем разделе приводится методика решения задач путем сведения к подзадачам с использованием графа редукции.

Четвертый раздел посвящен обзору и классификации моделей представления знаний. Определяются понятия «знания» и «база знаний», рассматриваются формальные и семантические модели представления знаний.

Пятый и шестой разделы отводятся под формальные системы - исчисление высказываний и исчисление предикатов. Авторы старались, насколько это достижимо в узких рамках пособия, дать по возможности полное, систематизированное и в то же время облегченное изложение этого трудного, но необходимого материала. Все изложенное сопровождается иллюстративными примерами и контрольными вопросами, что должно существенно облегчить понимание.

Этим, к сожалению, исчерпывается содержание данного пособия. Нельзя было и думать поместить в столь небольшое издание огромный материал, относящийся даже только к основам искусственного интеллекта. По этой причине авторы планируют продолжить издание пособий по ИИ.

Данное издание в первую очередь предназначено для студентов различных специальностей, включающих изучение ИИ. Оно также может быть полезно студентам, аспирантам, инженерам и научным работникам по всем другим специальностям, связанным с разработкой и применением автоматизированных информационных систем для любых сфер деятельности человека, в том числе и для гуманитарных.

1. ОСНОВНЫЕ ПОНЯТИЯ СИИ

1.1. Классификация задач, решаемых человеком

Наша жизнь постоянно вынуждает нас принимать самые разные решения, от простых (куда пойти, как пройти, когда, зачем и т.п.) до сложнейших. Как следует из Введения, развитие теории и практики систем искусственного интеллекта сопровождалось постановкой все более сложных "интеллектуальных" задач. При этом все задачи, решаемые человеком в своей практической и научной деятельности, можно подразделить на три основных класса.

1). Задачи, для которых были известны схемы решений и которые можно описать на некотором формальном языке. Примером таких задач служат все задачи, связанные с вычислениями (арифметика, алгебраические, дифференциальные, интегральные уравнения и системы и т.п.). Для таких задач всегда существует алгоритм решения, который обязательно приведет к искомому результату. Сюда же можно отнести задачи, для которых нет готовых схем решений, но они могут быть найдены или построены, если привлекаются дополнительные знания о данной предметной области (т.е. выделенной узкой сфере деятельности). К этому же классу относятся и задачи на адаптацию.

Именно для задач первого класса более всего были приспособлены первые ЭВМ фон-неймановской архитектуры. Общая схема их решения в самом простом виде выглядит так (рис.1.1)





Рис. 1.1. Схема решения задач на ЭВМ фон-неймановской архитектуры

На входе алгоритма - значения заданных переменных x_i , на выходе - значение выходных переменных y_j . Все вычисления в алгоритме основаны на 4-х арифметических действиях, как бы сложны они не были. Вычислительный характер действий и четкий алгоритм на много лет вперед определили путь развития ЭВМ и всех систем автоматизации. Были созданы соответствующие языки - АЛГОЛ, ФОРТРАН и др. Только после долгих колебаний и исследований задачи этого класса все-таки перестали относить к классу интеллектуальных.

2). Еще один класс задач, имеющих массовый характер, составляют задачи, методология решения которых определила целую эпоху в эволюции ЭВМ. Это задачи информационного поиска. Схема поиска решений здесь выглядит так (см. рис.1.2).

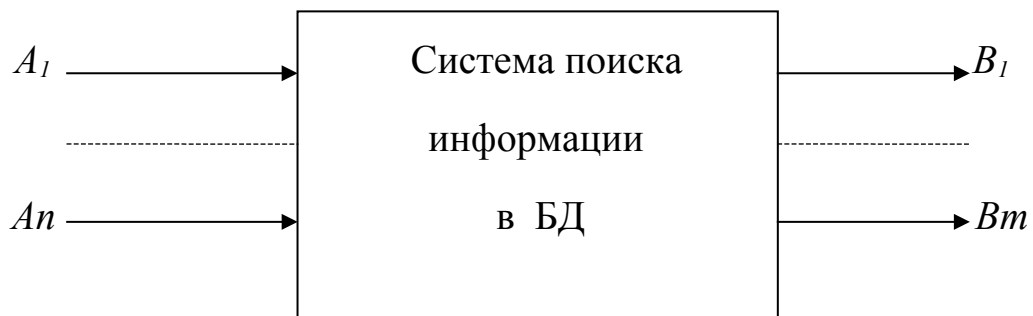


Рис. 1.2. Схема решения задач информационного поиска

На входе системы - запросы абонентов $A_1, A_2, \dots, A_n$. Решением задачи является некоторая часть базы данных (БД), соотносительная (релевантная) запросам - $B_1, B_2, \dots, B_m$. При этом информация, выдаваемая в качестве результата, не подвергается каким-либо изменениям. Она проходит лишь операции сортировки и некоторые другие теоретико-множественные операции над символьной информацией (т.е. информацией, представленной как в цифровой, так и в буквенной формах).

Переход к символьным задачам позволил впервые понять возможности разделения информации и механизмов ее обработки, т.е. отделить сами БД от алгоритмов их анализа и получения результатов. Это стало возможным благодаря заданию синтаксических схем организации записей в БД и ориентации алгоритмов только на синтаксическую обработку. Второй важной победой этого направления стало создание достаточно универсальных систем управления БД с широчайшей сферой практического применения.

3). Задачи этого класса характеризуются отсутствием априори известных схем решения, даже при условии привлечения дополнительных знаний о предметной области. Поэтому для решения таких задач используются сложные иерархические построения и программы, имитирующие механизмы мышления человека. К этому классу относятся, например, задачи планирования поведения объекта в сложных средах, задачи принятия решений в нестандартных (нештатных) ситуациях, задачи проектирования и конструирования, игры и т.п.. Этот класс задач принято считать интеллектуальными и именно для их решения разрабатывалась теория СИИ.

1.2. Основные понятия и определения

Любая наука отличается от других предметом своего исследования. В зависимости от предмета формулируются цели и задачи, а также методы и средства их решения. Если говорить о такой науке как искусственный интеллект, то в самом ее названии выделен объект исследования - интеллект, и основная ее задача - создание некоторого подобия «живого» интеллекта искусственными средствами. Искусственные средства известны были сразу - ЭВМ, включая всякие периферийные устройства, с помощью которых могут создаваться искусственные аналоги всего того, что может относиться к классу интеллектуального. Остается понять, что относить к классу интеллектуального, т.е. какие мыслительные процессы, выполняемые человеком, можно считать интеллектуальными, а какие нет.

Выше мы уже видели, что после долгих размышлений и колебаний в класс интеллектуальных, в конце концов, попали все задачи, алгоритм решения которых заранее не известен. Это и задачи распознавания образов, и машинный перевод с одного языка на другой, и принятие решений в зависимости от меняющихся ситуаций и многие другие. Все это стимулировало новые научные

исследования, целью которых было определить особенности интеллектуальной деятельности человека. Было показано, что в основе интеллектуального поведения человека лежит целый ряд метапроцедур, действие которых инвариантно относительно конкретно решаемых задач и областей деятельности.

В первую очередь, это метапроцедуры целенаправленного поиска, подобные целенаправленному поиску в лабиринте возможностей, ассоциативный поиск и ассоциативное рассуждение и др..

Поиск в лабиринте возможностей (лабиринтная гипотеза мышления) привела к появлению целого научного направления - эвристического программирования. Если лабиринт неизвестен, а должен порождаться, исходя из условий, конкретно формирующихся в данной предметной области, то необходимо определить такие метапроцедуры, которые позволили бы решить задачу. Так возникло новое научное направление - ситуационное моделирование (Поспелов Д.А., Клыков Ю.М., Загадская Л.С., Болотова Л.С.). Это направление выделило еще ряд метапроцедур: классификация ситуаций по признакам и по структурам, процедуры порождения новых классов понятий, отношений и ситуаций и т.п..

Следующим важным шагом в развитии ИИ было осознание необходимости внутреннего представления проблемной ситуации, что привело к выделению метапроцедур, оперирующих с совокупностью знаний из той предметной области, к которой принадлежит задача. Это так называемая модельная гипотеза. Основные метапроцедуры здесь: представление знаний, их пополнение и модификация, рассуждения, поиск релевантной (соответствующей) информации в совокупности имеющихся знаний и др..

Все перечисленные метапроцедуры в совокупности со знаниями о предметной области составили в конце концов ядро искусственного интеллекта. Теперь мы можем определить основные понятия, с которыми будем оперировать дальше.

Предметная область (ПО) – выделенная узкая сфера деятельности человека, относящаяся к данной задаче.

Модель предметной области (Мпо) – совокупность (система) знаний, необходимых для автоматического синтеза алгоритма решения задачи в данной области.

Искусственный интеллект (ИИ) (artificial intelligence - AI) – совокупность метапроцедур - представления знаний, рассуждений, поиска релевантной информации в среде имеющихся знаний, их пополнение, корректировка и т.п., - имитирующей деятельность человека.

Система искусственного интеллекта (СИИ) - аппаратный и информационно-программный комплекс, действие которого аналогично действию механизмов мышления человека и неотличимо от решений, которые принимались бы человеком-экспертом, т.е. профессионалом в данной предметной области.

Из этих определений следует, что любую систему ИИ отличают следующие особенности:

- наличие модели предметной области;
- наличие моделей механизмов мышления, т.е. метапроцедур, работающих на системе знаний, представленных моделью предметной области (в частности, процедур логического вывода);
- наличие естественно-языкового интерфейса, обеспечивающего взаимодействие пользователя с СИИ.

Очень часто под СИИ понимаются системы, основанные на знаниях. Раскрытию понятия знания мы отведем особое место.

1.3. Модель предметной области

Задача, предъявленная для решения, уже по своему заданию определяет круг объектов, действий, законов, справочных данных и т.п., необходимых для ее решения. Если задача из области медицины, то для ее решения, очевидно, бессмысленно закладывать (или запрашивать) данные о скорости расширения Галактики или о запасах угля в Подмосковном бассейне. Все, что нам нужно знать для решения задачи, как раз и составляет некую *предметную область* (ПО), которую мы формализуем в виде *модели предметной области* (Мпо) следующим образом:

$$Mpo : \langle X, C, R, G \rangle, \quad (1.1)$$

где обозначено:

$X = (x_1, x_2, \dots, x_n)$ – множество имен объектов (предметов, сущностей и т.п. внешнего мира), с которыми мы имеем дело при решении задачи.

$C = (c_1, c_2, \dots, c_m)$ – множество имен свойств (состояний) объектов, причем возможно, что $c_1 = (c_{11}, c_{12}, \dots, c_{1k})$, $c_2 = (c_{21}, c_{22}, \dots, c_{2r})$

и т.д. Эти свойства могут меняться под действием некоторых операторов.

$R = (r1, r2, \dots r_n)$ – множество имен отношений, в которые могут вступать объекты моделируемой ПО.

$G = (g1, g2, \dots g_k)$ – множество имен операций (действий), которые допустимы с этими объектами через изменение их свойств и отношений между ними.

Допустим, мы описываем предметную область задачи о сборе яблок. Тогда множество X – объекты: *ЯБЛОКО*, *ЛЕСТНИЦА*, *КОРЗИНА*, *ЯЩИК*;

множество C – свойства (состояния) объектов:

для *ЯБЛОКА* – *сорвано, уложено, отброшено*;

для *ЛЕСТНИЦЫ* – *лежит, стоит*;

для *КОРЗИНЫ* – *полна, пуста*;

для *ЯЩИКА* – *брошен, уложен*;

множество R – отношения:

ЯБЛОКО – *НА* ветке, *В* корзине;

ЛЕСТНИЦА – *У* дерева, *ПОД* деревом;

КОРЗИНА – *НА* лестнице, *У* ящика;

ЯЩИК – *У* дороги;

множество G – действия (операторы):

$g1$ – *поставить (ЛЕСТНИЦУ)*,

$g2$ – *повесить (КОРЗИНУ)*,

$g3$ – *сорвать (ЯБЛОКО)*,

$g4$ – *положить (ЯБЛОКО в корзину)*,

$g5$ – *наполнить (КОРЗИНУ)*,

$g6$ – *уложить (ЯЩИК)*,

$g7$ – *перенести (ЯЩИК к дороге)* и т.п.

Выделенные множества X , C , R , G позволяют представлять предметную область в виде некоторого языкового эквивалента, степень адекватности которого реальному миру зависит от множества факторов: сложности задач, числа задач, сложности самой ПО (число объектов, свойств, отношений, их семантика и т.д.).

Говорят, что множества X , C , R , G задают *концептуальную модель предметной области*. Они определяют ее статическую структуру. Для перехода к модели предметной области необходимо задать *пространство состояний*.

Говоря о модели ПО, мы отдаем себе отчет в том, что состояние ПО вообще-то непостоянно, оно может меняться со временем, и

поэтому правильнее было бы говорить о *состоянии предметной области* в данный момент времени.

Определим состояние ПО следующим образом:

$$Sno(t) : \langle X(t), C(t), R(t) \rangle. \quad (1.2)$$

Выражение (1.2) описывает ситуацию, сложившуюся в ПО в данный момент времени. В зависимости от этой ситуации на момент $t \in (t_{нач}, t_{кон})$ человек (или робот) выбирает из множества G возможных операций именно те действия, которые необходимы для решения задачи. Если теперь обозначить через F отображение $Sno(t)$ на множество G , то правомерно написать:

$$F : (\langle X(t), C(t), R(t) \rangle \rightarrow G). \quad (1.3)$$

Другими словами, сама предметная область выступает здесь в качестве пассивного элемента действия (объекта преобразования), а человек или система его заменяющая - в качестве преобразователя (субъекта действия). Таким образом отображается связь между языковым описанием ПО, ее состоянием (декларативная компонента), с одной стороны, и именами действий, выражающих процедурное знание (процедурная компонента), с другой. Состояние предметной области проектируется на множество действий.

1.4. Процедура решения задачи

Исходя из сказанного, определим теперь понятия "задача", "решение", "алгоритм".

Обозначим начальное состояние ПО через S_n . *Задача* заключается в том, чтобы перевести предметную область из состояния S_n в некоторое заданное, определяемое как целевое (S_u). Очевидно, сделать это возможно, лишь применяя допустимые в данной предметной области действия из множества $G = (g_1, g_2, \dots, g_k)$. Какие выбрать операции g_i и в какой последовательности - неизвестно. В этом как раз и состоит *решение задачи*. Схема решения, таким образом, выражается формулой:

$$Z = (S_n \xrightarrow{G} S_u). \quad (1.4)$$

Допустим, оператором g_1 , взятым из множества G , мы перевели состояние S_n в S_1 , но оно не совпало с S_u , т.е. $S_1 = g_1(S_n)$, $S_1 \neq S_u$. Для перевода S_1 в состояние S_2 применим оператор $g_2 \in G$: $S_2 = g_2(S_1)$ и, если S_2 снова не совпадет с S_u , обратимся к оператору g_3 , чтобы получить состояние S_3 и оценить его, сравнивая с S_u . И так далее, пока не найдется такой g_j , что $S_j = g_j(S_{j-1})$ и $S_j = S_u$. Описанный путь поиска решения можно отобразить следующей цепочкой:

$$S_n \xrightarrow{g_1} S_1 \xrightarrow{g_2} S_2 \xrightarrow{g_3} \dots \xrightarrow{g_{j-1}} S_j = S_u. \quad (1.5)$$

То же самое можно написать по-другому:

$$S_u = g_j(g_{j-1}(\dots(g_3(g_2(g_1(S_n)))))).$$

Последовательность $(g_1, g_2, g_3, \dots, g_j)$ как раз и будет представлять из себя *алгоритм* решения задачи.

Отметим, что перевод $S_n \rightarrow S_u$ возможен не единственным способом. В этом случае можно ставить задачу об оптимизации решения. Рассмотрим примеры построения модели предметной области и решения задачи.

1.5. Примеры решения задач

Пример 1. Рассмотрим широко известную задачу об обезьяне и бананах [7]. В комнате, где находится обезьяна, имеется ящик и связка бананов, причем бананы подвешены к потолку настолько высоко, что обезьяна не может дотянуться до них с пола. Какую последовательность действий должна совершить обезьяна, чтобы достать эти бананы? (Предполагается, что она должна догадаться подтащить ящик к бананам, взобраться на него и дотянуться до них).

Построим модель данной предметной области.

X - имена предметов и объектов Мпо: *ОБЕЗЬЯНА* (далее O), *ЯЩИК* ($Я$), *БАНАНЫ* ($Б$);

C - координаты O , $Я$, $Б$ в комнате - по горизонтали и вертикали. Значение координат - произвольное множество точек комнаты: $a, b, c, \dots$ (допустим, что это множество конечно).

R - имена отношений между объектами: *РЯДОМ*($O, Я$), т.е. обезьяна находится у ящика; *НА*($O, Я$) - обезьяна на ящике, наконец, *В*($O, Б$) - бананы в руках обезьяны. Очевидно, что возможны отрицания этих отношений: *НЕ-РЯДОМ*($O, Я$), *НЕ-В*($O, Б$), *НЕ-НА*($O, Я$) или иначе: $\overline{РЯДОМ}(O, Я)$, $\overline{В}(O, Б)$, $\overline{НА}(O, Я)$.

G - имена действий, которые может выполнить обезьяна:

ПОДОЙТИ ($O, Я$) = g_1 - обезьяне подойти к ящику;

ПЕРЕДВИНУТЬ ($O, Я$) = g_2 - обезьяне передвинуть ящик;

ВЗОБРАТЬСЯ ($O, Я$) = g_3 - обезьяне взобраться на ящик;

СХВАТИТЬ ($O, Б$) = g_4 - обезьяне схватить бананы.

ПЕРЕЙТИ (O, d) = g_5 - обезьяне перейти в точку с двумерной координатой $d(e, f, m, \dots)$.

Все действия имеют в качестве аргументов объекты Мпо, а субъект здесь единственный - обезьяна. Оператор g_5 отображает

возможные перемещения обезьяны по комнате без ящика. Пока он нам не пригодится, но в дальнейшем будет полезен.

Теперь нам необходимо построить отображение $F: (S(t) \rightarrow G)$, т.е. множества состояний на множество действий. С этой целью дадим описание начального и целевого состояний в Мпо. Допустим, что в S_n двумерная координата O (обезьяны) точка a , координата $Я$ - точка b , координата $Б$ - точка c , расположенная непосредственно под бананами. Тогда начальное состояние будет удобно описать в виде

$$S_n = (a, b, c, \overline{НА}(O, Я), \overline{В}(O, Б)), \quad (1.6)$$

все объекты в начальных позициях, обезьяна *НЕ-НА ЯЩИКЕ*, бананы *НЕ-В* руках у обезьяны. Каждое из отношений *НА* и *В* может принимать лишь два значения *ДА* ("1") и *НЕТ* ("0"): или обезьяна *НА* ящике ("1"), или *НЕТ* ("0"); или *В* руках у нее бананы ("1"), или *НЕТ* ("0"). Теперь мы можем упростить запись состояний (ср. с (1-6)):

$$S_n = (a, b, c, 0, 0), \quad (1.7)$$

$$S_u = (c, c, c, 1, 1). \quad (1.8)$$

(Все объекты в одной точке, обезьяна на ящике и схватила банан).

Задача, таким образом, сводится к тому, как с помощью вышеперечисленных операторов g_1, g_2, g_3, g_4 из ситуации (1.7) перейти в ситуацию (1.8). Применяем к S_n оператор g_1 - *ПОДОЙТИ*, имеем:

$$S_n = (a, b, c, 0, 0) \xrightarrow{g_1} (b, b, c, 0, 0) = S_1.$$

(Обезьяна *ПОДОШЛА* к ящику, и их координаты совпали).

Применяем g_2 - *ПЕРЕДВИНУТЬ*:

$$S_1 = (b, b, c, 0, 0) \xrightarrow{g_2} (c, c, c, 0, 0) = S_2.$$

g_3 - *ВЗОБРАТЬСЯ*:

$$S_2 = (c, c, c, 0, 0) \xrightarrow{g_3} (c, c, c, 1, 0) = S_3.$$

g_4 - *СХВАТИТЬ*:

$$S_3 = (c, c, c, 1, 0) \xrightarrow{g_4} (c, c, c, 1, 1) = S_u.$$

Весь процесс можно отобразить формулой

$$S_u = g_4(g_3(g_2(g_1(a, b, c, 0, 0)))). \quad (1.9)$$

Правила для описания ситуаций могут быть обобщены для случая, когда координаты нахождения Обезьяны и Ящика являются переменными. Обозначим их как x, y , соответственно. Координаты Бананов тоже можно сделать переменными, но пока, для упрощения ситуации, не будем этого делать. Тогда каждое правило будет описывать класс состояний, к которым применим тот или иной оператор. Эти правила будут выглядеть следующим образом:

$$(x,y,c, 0,0) \rightarrow g_1 \rightarrow (y,y,c,0,0) \quad (1.10)$$

$$(y,y,c, 0,0) \rightarrow g_2 \rightarrow (c,c,c,0,0) \quad (1.11)$$

$$(c,c,c, 0,0) \rightarrow g_3 \rightarrow (c,c,c,1,0) \quad (1.12)$$

$$(c,c,c, 1,0) \rightarrow g_4 \rightarrow (c,c,c,1,1) \quad (1.13)$$

Подсчитаем число состояний (мощность N) пространства состояний, описываемого каждым правилом .

Для (1.10): $N_1 = 20 \times 20 \times 1 \times 1 \times 1 = 400$,

для (1.11): $N_2 = 20 \times 1 \times 1 \times 1 \times 1 = 20$,

для (1.12): $N_3 = 1 \times 1 \times 1 \times 1 \times 1 = 1$,

для (1.13): $N_4 = 1 \times 1 \times 1 \times 1 \times 1 = 1$.

Здесь x, y определены на множестве точек комнаты $n = 20$, знак декартова произведения. А все пространство состояний имеет мощность, равную сумме мощностей подпространств, т.е. 422 состояния.

Данную модель можно сколько угодно усложнять: может меняться местоположение бананов, ящиков может быть несколько, в комнате могут находиться другие предметы и т.п., но смысл преобразований при этом не изменится.

Пример 2. О наполнении ведра водой.

В начальный момент времени пустое ведро стоит рядом с раковиной, кран закрыт. В целевой ситуации необходимо, чтобы ведро было полно и стояло на полу у раковины, а кран был бы закрыт. Все операции выполняет робот. Требуется построить план его действий.

Множество X : $(РОБОТ(P), ВЕДРО(V), КРАН(K), ПОЛ(П), РАКОВИНА(PK))$.

Множество C :

состояние $РОБОТА - (У КРАНА)$;

состояние $ВЕДРА - (ПУСТО, ПОЛНО)$;

состояние $КРАНА - (ОТКРЫТ, ЗАКРЫТ)$.

Множество R : $НА(V,П)$ - ведро на полу,

$В(V,PK)$ - ведро в раковине.

Поскольку робот все время находится в одной точке, его также можно исключить из рассмотрения. Тогда

$$S_H = \langle V(\text{ПУСТО}), K(\text{ЗАКРЫТ}), HA(V, П) \rangle.$$

$$S_U = \langle V(\text{ПОЛНО}), K(\text{ЗАКРЫТ}), HA(V, П) \rangle.$$

Рассмотрим состояния и действия.

1. $(V(\text{ПУСТО}), K(\text{ЗАКРЫТ}), HA(V, П)) \xrightarrow{g1} (V(\text{ПУСТО}), K(\text{ЗАКРЫТ}), \overline{HA}(V, П));$
2. $(V(\text{ПУСТО}), K(\text{ЗАКРЫТ}), \overline{HA}(V, П)) \xrightarrow{g2} (V(\text{ПУСТО}), K(\text{ОТКРЫТ}), \overline{HA}(V, П));$
3. $(V(\text{ПУСТО}), K(\text{ОТКРЫТ}), \overline{HA}(V, П)) \xrightarrow{g3} (V(\text{ПОЛНО}), K(\text{ОТКРЫТ}), \overline{HA}(V, П));$
4. $(V(\text{ПОЛНО}), K(\text{ОТКРЫТ}), \overline{HA}(V, П)) \xrightarrow{g5} (V(\text{ПОЛНО}), K(\text{ЗАКРЫТ}), \overline{HA}(V, П));$
5. $(V(\text{ПОЛНО}), K(\text{ЗАКРЫТ}), \overline{HA}(V, П)) \xrightarrow{g4} (V(\text{ПОЛНО}), K(\text{ЗАКРЫТ}), HA(V, П)).$

Последняя ситуация является, как видим, целевой.

Примечание 1. Из определения Мпо следует, что отображение F описывает условия применения операторов из множества

Примечание 2. Из примеров хорошо видно, как важно правильно (удачно) выбрать способ описания S_H и S_U . От этого зависят наглядность представления и даже скорость поиска решения.

1.6. Выводы

В пункте "Модель предметной области" были определены понятие "состояние МПО" и отображение F между множеством состояний и теми действиями, которые переводят предметную область из одного состояния в другое. Очевидно, что при этом необходимо указывать только те действия (операторы), которые могут изменить состояние Мпо путем изменения его описания. - Например, в задаче об обезьяне и бананах действие ("залезть") может быть применено только к состоянию с описанием $g^3(c, c, c, 0, 0)$, а в результате будет получено новое состояние - $(c, c, c, 1, 0)$ как результат выполнения этого действия. Так же, по действиям, могут быть найдены и остальные состояния (прототипы состояний). Применение операторов к описанию начального состояния, а затем и ко всем промежуточным состояниям, порождает все множество возможных

состояний предметной области, называемое *пространством состояний*.

В связи с этим возникает ряд новых вопросов: как строить это пространство, чтобы не потерять ни одного состояния, как ориентироваться в этом пространстве для поиска целевых состояний и т.п.. Эти вопросы как раз и рассматриваются в следующем разделе.

Вопросы для самопроверки

1. Как менялось понятие «задачи» на различных этапах развития представлений об интеллектуальной системе?
2. Что является объектом исследования в СИИ?
3. Какими особенностями обладают СИИ по сравнению с традиционными алгоритмическими системами?
4. Какими свойствами должна обладать любая интеллектуальная система?
5. Дайте определение СИИ.
6. Что понимается под моделью предметной области?
7. Какова структура предметной области?
8. Приведите примеры предметной области.
9. Постройте предметную область для близкой вам области деятельности (ремонт ЭВМ, утюга, автомобиля, подготовка щей, поход в магазин и т.п.).
10. Что мы понимаем под «решением» задачи?
11. Что такое алгоритм решения задач?
12. Требуется построить биссектрису угла треугольника.

Определите начальное и конечное (целевое) состояния предметной области и постройте пространство состояний для данной задачи.

13. Гипотенуза прямоугольного треугольника равна 5. Некий автомат, который может работать только с целыми числами, должен вычислить величины катетов. Постройте пространство состояний автомата.