ББК 32.813
С 34
УДК 007.32

Рецензенты: к.т.н. А.М. Варюха
к.т.н. Б.Е. Демин

С 34 Болотова Л.С., Смирнов Н.А., Смольянинов А.А. Системы искусственного интеллекта. Теоретические основы СИИ и формальные модели представления знаний: Учебное пособие/ Московский государственный институт радиотехники, электроники и автоматики (технический университет). - М., 2003. – 120 с.

ISBN 5-7339-0389-9

Пособие представляет собою систематизированное изложение основ теории интеллектуальных систем (ИИ). Даются основные понятия и определения. Приводится методика построения модели предметной области и процедура поиска решений в пространстве состояний. По возможности просто, но достаточно полно, излагаются аксиоматические системы – исчисление высказываний и исчисление предикатов. Материал иллюстрируется многочисленными примерами и содержит вопросы для самопроверки и упражнений.

Пособие предназначено для студентов и аспирантов специальностей, связанных с изучением ИИ, а также для всех интересующихся вопросами построения и применения СИИ.

Табл. 3. Илл. 18. Библиогр.: 29 назв.

Печатается по решению редакционно-издательского совета Московского государственного института радиотехники, электроники и автоматики (технического университета).

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 (1V 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 перемешивались 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 обозначались 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., Гелентер 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 кон струкций), 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, со вмещенного 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 недосягаемыми для других, пока еще только развивающихся, стран. Именно ИИТ сегодня в мире являются источником глобальной власти и материального благополучия.

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 облегчённое 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)

На входе алгоритма - значения заданных переменных x_i, на выходе - значение выходных переменных *yj*. Все вычисления в алгоритме основаны на 4-х арифметических действиях, как он 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.

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

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).

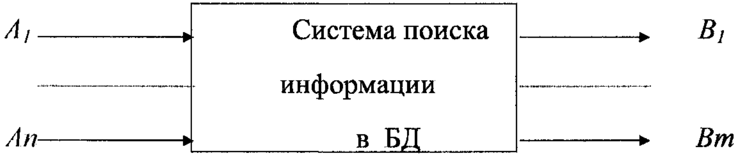


Рис.1.2. Схема решения задач информационного поиска

На входе системы - запросы абонентов A_1, A_2,.. An. Решением задачи является некоторая часть базы данных (БД), соотносительная (релевантная) запросам - B_1, B_2,.. Bm. 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 в 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., - имитирующих 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:

$$Mno:\ \langle X, C, R, G \rangle, \tag{1.1}$$

где обозначено:

$X = (x_1, x_2, \ldots x_n)$ – 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 некоторы операторов.

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

$G = (g1, g2, \dots gk)$ - множество имен операций (действий) которые допустимы с этими объектами через изменение и 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 представлят 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:

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

Выражение (1.2) описывает ситуацию, сложившуюся в ПО в данный момент времени. В зависимости от этой ситуации на момент $t \in (tнач, tкон)$ 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:

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

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 $Sн$. *Задача* заключается в том, чтобы перевести предметную область из состояния $Sн$ в некоторое заданное, определяемое как целевое ($Sц$). Очевидно, сделать это возможно, лишь применяя допустимые в данной предметной области действия из множества $G = (g1, g2, \dots gk)$. Какие выбрать операции gi и в какой последовательности – неизвестно. В этом как раз и состоит *решение задачи.* Схема решения, таким образом, выражается формулой:

$$Z = (Sн \xrightarrow{G} Sц). \qquad (1.4)$$

Допустим, оператором $g1$, взятым из множества G, мы перевели состояние $Sн$ в S_1, но оно не совпало с $S_Ц$, т.е. $S_1 = g_1(Sн)$, $S_1 \neq Sц$. 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 $Sц$, обратимся к оператору g_3, чтобы получить состояние S_3 и оценить его, сравнивая с $Sц$. И так далее, пока не найдется такой gj, что $Sj = gj(Sj_{-1})$ и $Sj = Sц$. Описанный путь поиска решения можно отобразить следующей цепочкой:

$$Sн \xrightarrow{g_1} S_1 \xrightarrow{g_2} S_2 \xrightarrow{g_3} \ldots \xrightarrow{g_{j-1}} Sj = Sц. \qquad (1.5)$$

То же самое можно написать по-другому:

$$Sц = gj(gj_{-1}(\ldots(g_3(g_2(g_1(Sн)))\ldots))).$$

Последовательность (g_1, g_2, g_3, ... gj) как раз и будет представлять из себя *алгоритм* решения задачи.

Отметим, что перевод $Sн \rightarrow Sц$ 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 *О*), *ЯЩИК* (*Я*), *БАНАНЫ* (*Б*);

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

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

G - имена действий, которые может выполнить обезьяна:

ПОДОЙТИ (О,Я) = *g1* - обезьяне подойти к ящику;

ПЕРЕДВИНУТЬ (О, Я) = *g2* - обезьяне передвинуть ящик;

ВЗОБРАТЬСЯ (О, Я) = *g3* - обезьяне взобраться на ящик;

СХВАТИТЬ (О, Б) = *g4* - обезьяне схватить бананы.

ПЕРЕЙТИ (О, d) = *g5* - обезьяне перейти в точку с двумерной координатой *d* (*e*, *f*, *m*...).

Все действия имеют в качестве аргументов объекты Мпо, а субъект здесь единственный - обезьяна. Оператор *g5* 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 *Sн* двумерная координата *О* (обезьяны) точка *a*, координата *Я* - точка *b*, координата *Б* - точка *c*, расположенная непосредственно под бананами. Тогда начальное состояние будет удобно описать в виде:

$$Sн = (a, b, c, \overline{НА}(О, Я), \overline{В}(О, Б)), \qquad (1.6)$$

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)):

$$Sн = (a, b, c, 0, 0), \qquad (1.7)$$

$$Sц = (c, c, c, 1, 1). \qquad (1.8)$$

(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 *Sн* оператор g_1 - *ПОДОЙТИ*, имеем:

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

(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*:

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

Весь процесс можно отобразить формулой:

$$S_{ц} = g_4(g_3(g_2(g_1(a, b, c, 0, 0)))). \qquad (1.9)$$

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:

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

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

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

$$(c,c,c,1,0) \rightarrow g_4 \rightarrow (c,c,c,1,1)\ . \qquad (1.13)$$

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 $n = 20$, [illegible] декартова произведения. А все пространство состояний имеет мощность, равную сумме мощностей подпространств, т.е. 422 состояния.

Данную модель можно сколько угодно усложнять: может меняться и 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*(*Р*), *ВЕДРО*(*В*), *КРАН*(*К*), *ПОЛ*(*П*), *РАКОВИНА*(*РК*)).

Множество *С*:

состояние *РОБОТА* – (*У КРАНА*);
состояние *ВЕДРА* – (*ПУСТО, ПОЛНО*);
состояние *КРАНА* – (*ОТКРЫТ*, *ЗАКРЫТ*).

Множество *R*: *НА*(*В,П*) - ведро на полу,
В(*В,РК*) - ведро в раковине.

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

$S_н$ = < *В*(*ПУСТО*), *КРАН*(*ЗАКРЫТ*), *НА*(*В,П*) >.
$S_ц$ = < *В*(*ПОЛНО*), *КРАН*(*ЗАКРЫТ*), *НА*(*В,П*) >.

Рассмотрим состояния и действия.

1. (*В* (*ПУСТО*), *К* (*ЗАКРЫТ*), *НА* (*В,П*)) $\longrightarrow g1 \longrightarrow$
$\longrightarrow$ (*В* (*ПУСТО*), *К* (*ЗАКРЫТ*), $\overline{НА}$ (*В,П*));
2. (*В* (*ПУСТО*), *К* (*ЗАКРЫТ*), $\overline{НА}$ (*В,П*)) $\longrightarrow$ g2 $\longrightarrow$
$\longrightarrow$ (*В* (*ПУСТО*), *К* (*ОТКРЫТ*), $\overline{НА}$ (*В,П*));
3. (*В* (*ПУСТО*), *К* (*ОТКРЫТ*), $\overline{НА}$ (*В,П*)) $\longrightarrow g3 \longrightarrow$
$\longrightarrow$ (*В* (*ПОЛНО*), К (*ОТКРЫТ*), $\overline{НА}$ (*В,П*));
4. (*В* (*ПОЛНО*), *К* (*ОТКРЫТ*), $\overline{НА}$ (*В,П*)) $\longrightarrow g5 \longrightarrow$
$\longrightarrow$ (*В* (*ПОЛНО*), *К* (*ЗАКРЫТ*), $\overline{НА}$ (*В,П*));
5. (*В* (*ПОЛНО*), *К* (*ЗАКРЫТ*), $\overline{НА}$ (*В,П*)) $\longrightarrow g4 \longrightarrow$
$\longrightarrow$ (*В* (*ПОЛНО*), *К* (*ЗАКРЫТ*), *НА* (*В,П*)).

Последняя ситуация является, как видим, целевой.

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

Примечание 2. Из примеров хорошо видно, как важно пра вильно (удачно) выбрать способ описания *Sн* и *Sц*. 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 *g3* *(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 состояния 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.

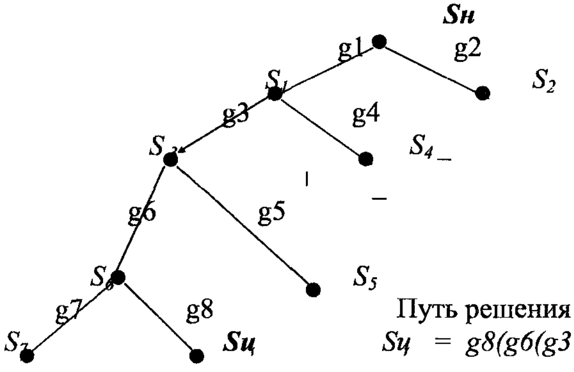
2. МЕТОДЫ ПОИСКА РЕШЕНИЙ В ПРОСТРАНСТВЕ СОСТОЯНИЙ

2.1. Путь решения задачи

Из предыдущих примеров мы видели, что применение операторов к начальному и другим, промежуточным, состояниям меняет эти состояния, порождая новые, которых вообще-то может быть много и которые образуют то, что называется *пространством состояний*. Среди этих состояний, если модель построена корректно, рано или поздно появится одно, которое будет соответствовать целевому, и процесс поиска решения на этом будет закончен. Процесс этот может быть длинным и сложным, поэтому возникает необходимость в некотором едином методе представления множества состояний и поиска решений. Таким методом является метод, удобной графической моделью которого стал граф.

Если отождествить состояние *Sн* с корнем или начальной вершиной графа-дерева, то, применяя к *Sн* какой-либо оператор $g1 \in G$, мы порождаем новое состояние S_1, образуя тем самым следующую вершину графа (рис. 2.1).

Эта новая вершина может быть промежуточной или целевой. Если вершина промежуточная, то процесс порождения новых вершин (с помощью операторов *gi*) будет продолжен, пока не найдется целевая. Процесс применения оператора *gi* к некоторой вершине называется *раскрытием вершины.* От каждой порожденной верш ны к породившей ее расставляются указатели, которые позволяю найти путь назад, к начальной вершине, после того, как обнаруж на целевая. Общая процедура построения дерева в пространст состояний при этом выглядит следующим образом.



Путь решения задачи:
$S_ц = g8(g6(g3(g1(S_н))))$

Рис. 2.1. Граф решения задачи

1) К корню дерева (*Sн*) применяются операторы *gi* из множе ства *G* (их может быть несколько). Полученные при этом верши ны образуют первый уровень новых вершин.

2) Каждая из вновь полученных вершин проверяется, не я вляется ли она целевой. Если нет, то процесс продолжается по о ношению к каждой из них. Образуется второй уровень верши Если к какой-либо вершине никакой оператор из *G* не примени то эта вершина становится терминальной (конечной). Как види на каждом шаге проводятся две операции: порождение ново вершины и проверка, не является ли новая вершина целевой, т. совпадающей с целевым состоянием.

3) Когда целевая вершина найдена, в обратном направлени (от цели к началу) просматриваются указатели дуг и выделяетс путь решения. Практически этот путь удобнее отображать п средством операторов, связанных с этими дугами (см. рис. 2.1).

В общем случае количество вершин может быть больши Их последовательное раскрытие, анализ и пометка пути ослож

ют задачу. Возникает проблема перебора вершин: в каком по- ядке они будут порождаться и анализироваться. Здесь возможны ледующие варианты:

1) вершины раскрываются в том же порядке, в котором они орождаются, то такой перебор называется полным перебором в ширину (*breadth - first process*);

2) на каждом шаге первой раскрывается вершина, которая ыла построена последней, такой процесс называется полным пе- ебором в глубину (*depth - first process*). В этих процессах распо- ожение целевой вершины не влияет на порядок раскрытия, по- тому их часто называют процессами слепого перебора.

3) если есть некоторая дополнительная (эвристическая) ин- формация о предметной области, которая позволяет делать сужде- ния о характере графа пространства состояний и расположения це- и, то такой метод построения графа называется эвристическим "эвристический" означает "служащий открытию"). Эвристическая нформация, опирающаяся, как правило, на предыдущий опыт, по- золяет выполнять поиск в наиболее перспективных направлениях.

Говоря о графе, будем рассматривать только один наиболее ростой его тип - граф типа "дерево". Как известно, деревом называ- тся граф, каждая вершина которого имеет только одну вершину, непосредственно предшествующую ей (родительскую), за исключе- нием вершины-корня, которая предшествующих вершин не имеет.

2.2. Метод полного перебора в ширину

Как уже было сказано, в этом методе вершины раскрываются том порядке, в котором они строятся. Основной алгоритм со- тоит в выполнении следующих действий.

1) Раскрывается начальная вершина *Sн*. Она раскрывается до ех пор, пока ее можно раскрыть, применяя один и тот же оператор или разные, смотря по условию). При этом образуются вершины ервого уровня: S_1, S_2, S_3...Они раскрываются в свою очередь, и об- разуются вершины второго уровня и т.д. (рисунок 2.1 может слу- жить примером этого метода: S_1 и S_2 - вершины первого уровня, S_3 и S_4 - вершины второго уровня, S_5 и S_6 - третьего и т.д.).

2) Расставляются указатели, ведущие от новых вершин к

корню. Это могут быть условные имена, буквы, цифры, имен операторов и т.п.. Но могут быть и реальные величины, например, расстояния, стоимость, вес и т.д.

3) Проверяется, нет ли среди полученных вершин целевой. Если есть, то формируется решение на основе соответствующег оператора. Если целевых вершин нет, то рассматривается перва порожденная вершина и к ней применяется тот же алгоритм. После чего, переходят ко второй и т.д., пока среди получаемых вершин не окажется целевой.

Метод полного перебора в ширину гарантируют нахождени целевой вершины как раз потому, что перебор - полный. Путе достижения цели, вообще говоря, может быть много. В этом случае у нас имеется возможность выбрать наикратчайший (или самый дешевый, или самый легкий - критериев много) путь. Н может быть случай, когда граф поиска окажется бесконечным тогда этот алгоритм никогда не кончит работу.

Таким образом, метод полного перебора гарантирует поис оптимального решения, если дерево пространства состояний н бесконечно.

Синонимами названия метода являются: метод грубой силы метод проб и ошибок.

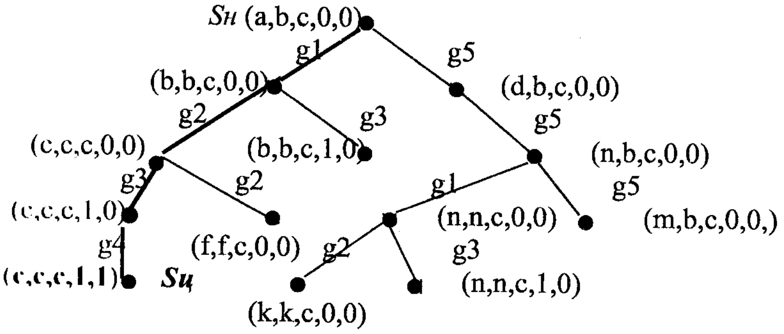


Рис. 2.2. Элемент дерева полного перебора в ширину для примера с обезьяной и бананами

На рис. 2.2 показан элемент дерева полного перебора в ширину для примера с обезьяной и бананами (для полноты изложения здесь применяется дополнительный оператор *g5*, отображающий перемещение обезьяны по комнате из точки в точку).

Корень дерева совпадает с *Sн*(*a*,*b*,*c*,0,0,). Точки *d*, *f*, *k*, *n*, *m* - координаты возможной миграции обезьяны по комнате. Таких точек, конечно, множество, но мы выбрали лишь несколько для примера. Они не приводят к решению, но теоретически вполне допустимы. Жирной линией показан путь до целевой вершины *Sц*(*c*,*c*,*c*,1,1), которому соответствует последовательность операторов *g4*, *g3*, *g2*, *g1*. Алгоритм решения отобразится формулой:

$$Sц = g4(g3(g2(g1(a,b,c,0,0)))).$$

2.3. Метод полного перебора в глубину

В отличие от метода перебора в ширину этот метод предлагает раскрывать прежде всего те вершины, которые были построены последними. Первой раскрываемой вершиной, а следовательно, и последней, является корневая, но процесс всегда будет идти по самой левой ветви вершин. Чтобы как-то ограничить перебор, вводится понятие глубины вершины в дереве перебора. Полагаем, что глубина корня дерева равна нулю, а глубина любой последующей вершины равна единице плюс глубина вершины, непосредственно ей предшествующей.

Отсюда следует, что наибольшую глубину всегда будет иметь та вершина, которая должна быть в этот момент раскрыта. Если образующийся путь оказывается бесполезным, то есть при заданной глубине раскрытия целевой вершины не получилось, необходимо вернуться в вершину, предшествующую раскрытой и попытаться еще раз применить к ней операцию раскрытия. И так до тех пор, пока не будет получена целевая вершина.

Возврат осуществляется с помощью указателей. Как только в процессе порождения вершин достигается заданная граничная глубина, раскрывается вершина наибольшей глубины, не превышающая этой границы. Общая схема перебора в глубину показана на рис. 2.3.

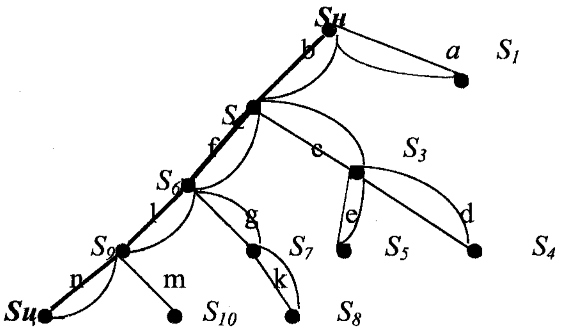


Рис. 2.3. Дерево полного перебора в глубину

Алгоритм перебора в глубину состоит в следующем.

1) Раскрывается начальная вершина, оответствующая начальному состоянию *Sн*.

2) Раскрывается первая вершина, получаемая в результат раскрытия *Sн*. Ставится указатель.

3) Если она раскрывается, то следующей будет раскрываться вновь порожденная вершина. Если вершина не раскрывается то процесс возвращается в предыдущую вершину.

4) По получении целевой вершины процесс раскрытия заканчивается и по указателям строится путь, ведущий к корню Соответствующие дугам операторы образуют решение задачи.

5) Если для заданной глубины раскрытия целевая вершин не находится, то весь процесс повторяется снова, а в качестве новой вершины рассматривается самая левая из полученных н предыдущем этапе.

Так же, как и метод поиска "в ширину", этот метод относится к методам "грубой силы". Он обеспечивает перебор всех состояний, если, конечно, прежде не встретит целевое.

2.4.Эвристические методы поиска в пространстве состояний

Методы полного перебора гарантируют решение задачи, если оно существует, а при наличии нескольких решений, гарантирует оптимальное. Однако на практике эти методы используютс

для решения лишь небольших по размерности графа состояний задач. Для реальных случаев чаще всего используется дополнительная информация, основанная на предыдущем опыте или полученная на основании теоретических выводов.

Такая информация называется *эвристической*, а организованная в правила - *эвристическими правилами* или *эвристиками*. Эвристическая информация носит сугубо специальный характер и может применяться только в рамках данной задачи, в лучшем случае в рамках задач данного класса.

Другими словами, эвристическая информация превращает грубый перебор в упорядоченный. В качестве примера рассмотрим известную задачу о коммивояжере.

Коммивояжер должен построить свой маршрут так, чтобы побывать в каждом из *n* городов в точности по разу и возвратиться в исходный город. Желательно также, чтобы этот маршрут был минимальным по протяженности. Пусть *n*=5. Города обозначим через *A,B,C,D,E*. Длина пути от города до города задана на рис. 2.4.

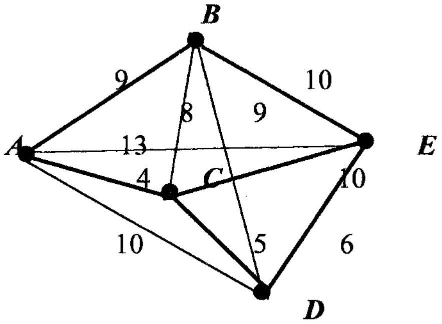


Рис. 2.4. Задача о коммивояжере

Пусть начальным пунктом отправления будет город *A*, поэтому *Sн* *A*. Целевое состояние - тоже *A*. Оператор перехода - единственный - g_1. Граф пространства состояний этой задачи при использовании метода поиска в ширину имеет вид, показанный на рис. 2.5.

Граф полного перебора, включающий все возможные пути коммивояжера, будет содержать 24 варианта. (На рис. 2.5 показаны лишь некоторые из них). Очевидно, среди этих путей будет и самый короткий (он выделен), и самый длинный. Расстояния километрах обозначены на каждом из отрезков. Из рисунка ви но, что длина оптимального пути - 34 км (*A,B,E,D,C,A*) и это путь единственный, если не считать того, что он может был пройден в обратном порядке *(A,C,D,E,B,A)*. (Кстати, если исклю чить «обратные» пути, вариантов будет всего 12).

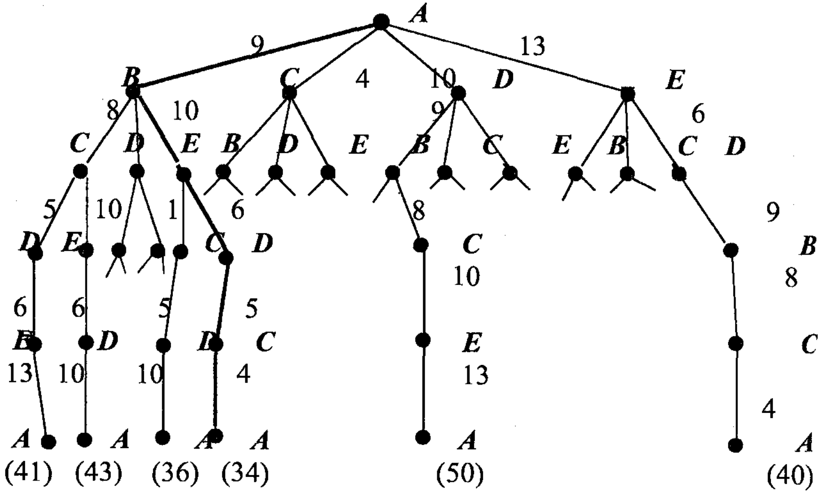


Рис. 2.5. Граф полного перебора в ширину для задачи о комм вояжере

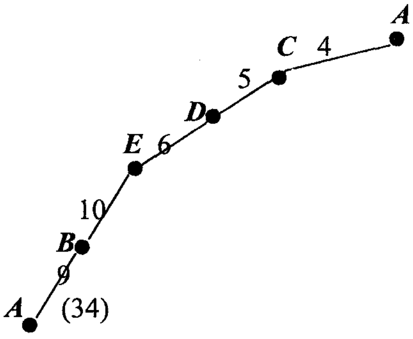


Рис. 2.6. Граф поиска решения задачи в глубину

Теперь построим решение методом поиска в глубину с ис-ользованием, например, такой эвристики: на каждом шаге пер-ой раскрывается вершина, имеющая самую короткую длину из ех возможных. Тогда граф поиска решений будет таким, как оказано на рис.2.6.

Существует много методов применения эвристических ценочных функций [4]. Процедура перебора на заданную убину часто называется также программированием с обратным ежением (*back - track programming*).

Возможно также использование комбинированных стратегий иска методами и в ширину, и в глубину: перебор этапами, ог-ничение числа дочерних вершин, двунаправленный перебор от и к *Sц* и обратно, методы ветвей и границ и динамическое про-раммирование, применяемые в исследовании операций.

Вопросы для самопроверки

1. Что представляет собой пространство состояний? Опреде-ите понятия начального и целевого состояний.
2. В какой форме можно представить пространство состояний?
3. Какие методы поиска в пространстве состояний вы знаете?
4. Как искать решение задачи по методу полного перебора в ирину?
5. Дайте определение решения задачи на дереве пространст-состояний.
6. Как определяется решение задачи по методу полного ребора в глубину?
7. Поясните алгоритм построения графа-дерева.
8. Какая информация называется эвристической?
9. В чем состоят эвристические методы перебора?
10. В чем заключаются достоинства и недостатки методов рубой силы»?
11. Имеются три бочонка объемом в 8, 5 и 3 литров. Первый полнен квасом, два остальных пустые. Требуется квас разде-ить пополам. Построить граф решения задачи по методу полно-о перебора в ширину.

Указание: Если вновь открытая вершина совпадает с от крытой ранее, она считается конечной.

12. То же – по методу полного перебора в глубину.

3. РЕШЕНИЕ ЗАДАЧ МЕТОДОМ РАЗБИЕНИЯ НА ПОДЗАДАЧИ

Идея метода состоит в следующем. Если путь решения зад чи неясен (или неизвестен), то нужно постараться найти какую- точку опоры, от которой (или до нее) путь известен. Это точ может быть в любом месте - в начале, в середине, в конце, но о должна быть. Этой точке, очевидно, будет соответствовать нек торое состояние ***Si*** на пути от *Sн* до *Sц*. Если эта точка *Si* опред лена, то к ней уже можно применить какое-либо действие и эт изменить состояние предметной области, т.е. приблизиться к во можному решению. Для нового состояния также ищется очка оп ры на оставшемся пути, к ней снова применяется какое-то (или же самое) действие и т.д., пока не будет решена вся задача.

Чтобы детально описать этот метод, необходимо вернуть к определению понятия задачи. Для этой цели воспользуемся п нятием пространства состояний. В более общем, чем прежде, в де задачу (*Z*) можно представить следующим образом:

$$Z = \langle \; S, G, F \; \rangle, \qquad (3.$$

где *S* - множество начальных состояний, *G* - множество операт ров, переводящих предметную область из одного состояния другое, *F* -множество целевых состояний.

При таком обозначении промежуточные состояния *Si* удо нее обозначать через *fi*, поскольку они теперь представляют с бой как бы промежуточные цели.

Конечной целью сведения задачи к подзадачам является п лучение таких элементарных задач, решение которых очевидн *Элементарными* считаются задачи, которые могут быть решен за один шаг, т.е. за одно применение какого-либо оператора множества *G*.

3.1. Представление задачи в виде И/ИЛИ графа

Между полученными при разбиении подзадачами мог быть отношения согласованности (одновременности) их решен

тношение ***И***), или отношение альтернативности (отношение ***ИЛИ***). Рассмотрим это подробнее.

Допустим, что исходная задача *Z* может быть разбита на подзадачи *A*, *B*, *C*, *D*, *E*. Пусть при этом подзадачи *A* и *B*, а также *C* и находятся в отношениях ***И***, т.е. должны решаться согласованно дновременно), а множества (*A* ***И*** *B*), (*C* ***И*** *D*) и подзадача *E* находится в отношении ***ИЛИ***, т.е. имеют альтернативный характер отношений. Тогда все сказанное легко отобразить на графе (рис.3.1).

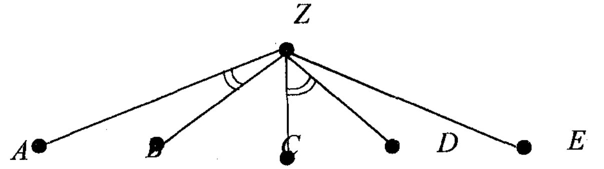


Рис. 3.1. Граф отношений подзадач для задачи Z

Отношения типа ***И*** будем отмечать двойной дугой, связывающей ребра графа. Для единообразия представления можно ввести дополнительные (фиктивные) вершины, которые группировали бы подзадачи типа ***И*** и ***ИЛИ*** под своей собственной *родительской* вершиной (рис. 3.2).

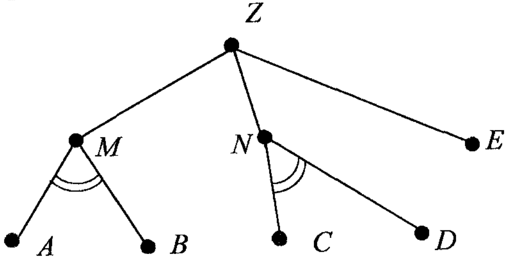


Рис. 3.2. И/ИЛИ граф задачи Z

Таким образом, исходная задача *Z* представляется подзада ами *M*, *N* и *E*, имеющими альтернативный *ИЛИ* - характер, а подзадачи *M* и *N*, в свою очередь, - подзадачами (*A*,*B*) и (*C*,*D*) с отношениями типа *И*. Альтернативные вершины *M*, *N*, *E* назы ются поэтому *ИЛИ* -вершинами, а вершины *A*, *B*, *C*, *D* - *И* вершинами и помечаются двойной дугой.

Структура типа изображенной на рис. 3.2 называется *И/ИЛ* графом. При этом, как видим, если вершина *И/ИЛИ* графа име непосредственно следующие за ней вершины, то либо все они я ляются *ИЛИ* - вершинами, либо вершинами *И*. Если у вершин имеется только одна следующая за ней вершина, то ее можно сч тать как *И*, так и *ИЛИ* - вершиной. Если вершин типа *И* вообі нет, то получается обычный граф, тот самый, который получался методе полного перебора при поиске в пространстве состояний.

Структура типа *И/ИЛИ* графа является удобной для пре ставления дерева подзадач. При этом начальная вершина граф соответствует начальному состоянию *Sн* исходной задачи, а ве шины, которые соответствуют описаниям элементарных подз дач, называются *заключительными*.

Основная цель поиска на *И/ИЛИ* графе - показать *разреш мость* вершины *Si*. Вершина является *разрешимой*, если выпол няется одно из следующих условий:

1) вершина *Si* является заключительной;

2) следующие за *Si* вершины являются вершинами тип *ИЛИ* и при этом хотя бы одна из них разрешима;

3) следующие за *Si* вершины являются вершинами типа и при этом каждая из них разрешима.

Решающим графом называется подграф, состоящий из ра решимых вершин с корнем в начальной вершине.

В случае,если у вершины *И/ИЛИ* графа, не являющейся з ключительной, нет следующих за ней вершин, такая вершина на зывается *неразрешимой*.

Порождение новых вершин (а эта операция еще называетс операцией *редукции задачи*) выполняется путем применени обобщенного оператора *G* (т.е. каких-либо операторов из множе ства *G*) сведения задачи к подзадачам. Применение оператора *G* описанию задачи порождает всю структуру *И/ИЛИ* графа (ил иначе *графа редукции*).

3.2. Механизм сведения задачи к подзадачам

Напомним, что наша цель - построить некоторый алгоритм который, будучи примененным чисто механически, обязательн [illegible] результат, т.е. требуемую структуру - *И/ИЛИ* граф с выде

ным решающим подграфом. Такой алгоритм представляет со й один из механизмов планирования решения задачи.

Как следует из предыдущего, если исходная задача *Z* зада я своим начальным описанием:

$$Z = \langle\ S,G,F\ \rangle,$$

свести ее к совокупности более простых задач в пространстве тояний можно, если нам удастся выделить основные проме точные состояния $f1, f2, f3...fn$. Каждому из этих состояний в ответствие можно поставить свое описание в виде троек:

$$\langle\ S, G, f1\ \rangle, \langle\ f1, G, f2\ \rangle, \langle\ f2, G, f3\ \rangle, ... \langle\ fn, G, F\ \rangle.$$

Решение этих подзадач эквивалентно решению исходной ачи. На этой идее построен основной механизм сведения зада к подзадачам.

1. Выделяем по крайней мере один оператор $gi \in G$, кото й обязательно будет участвовать в решающей цепочке опера ров. Все операторы такого типа называются *ключевыми* (т.е. ераторы, обязательно участвующие в решающей цепочке).

2. Для каждого из ключевых операторов (если их несколько) ределяются промежуточные состояния, к которым они могут ы применены в условиях задачи *Z*. Для оператора *gi* это будет тояние *fi*. (Таких состояний, вообще говоря, может быть не лько, и тогда они образуют подмножество целевых состояний *F*). Теперь уже можно выделить подзадачу поиска пути от ала до состояния *fi* (или до *Fgi*). Другими словами, примене оператора *gi* привело к первой подзадаче с описанием:

$$\langle\ S, G, fi\ \rangle \text{ (или } \langle\ S, G, Fgi\ \rangle).$$

3. Как только такое описание найдено, может быть сформу рована 2-я подзадача, соответствующая элементарной. В самом е, поскольку состояние *fi* соответствует оператору *gi*, то ни не мешает нам применить его к *fi* и получить, таким образом, ное состояние *gi(fi)*, определенно приближающее нас к конеч й цели, правда, только на один шаг.

4. От вновь полученного состояния *gi(fi)* до конечной цели ще может быть неблизкий путь. Определить его – третья под ача. Таким образом, применение выбранного оператора *gi* к

задаче с описанием $Z = \langle\ S, G, F\ \rangle$ позволяет выделить сразу т подзадачи: $\langle\ S, G, fi\ \rangle, \langle\ fi, gi, gi(fi)\ \rangle, \langle\ gi(fi), G, F\ \rangle$, одна из которых элементарная. Сказанное удобно проиллюс рировать на линейном графике (рис. 3.3). Все решение изображ ется отрезком, который разбивается точкой *fi* , соответствующ оператору *gi* на подзадачи.

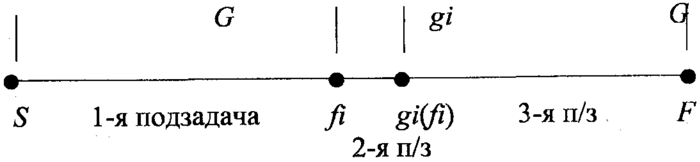


Рис. 3.3. Разбиение задачи на подзадачи (п/з)

Такому разбиению будет соответствовать следующ И/ИЛИ граф (рис.3.4).

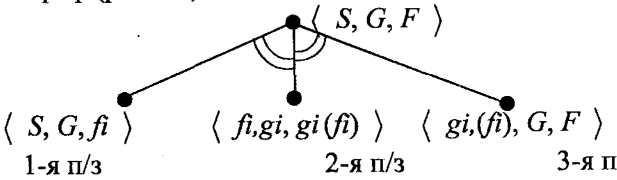


Рис. 3.4. И/ИЛИ граф разбиения на подзадачи для сост ния fi

Элементарная подзадача типа 2 решается всегда для люб выбранной точки *f* пространства состояния, поэтому ее можно указывать. Точка *fi* - одна из возможных промежуточных цел $fi \in Fgi$. Выбрав ее, мы применяем к ней оператор *gi*. Обобш сказанное, приходим к окончательному виду *И/ИЛИ* графа р биения на подзадачи для одной точки (рис.3.5).

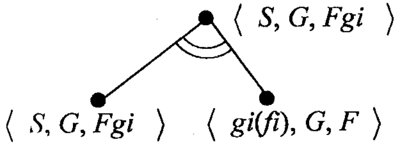


Рис. 3.5. Обобщенный И/ИЛИ граф для одной точки (fi)

5. Каждая из полученных при разбиении подзадач, если она, онечно, не элементарная, может быть снова разбита на подзадачи налогичным методом, возможно, с помощью другого оператора.

Примечание. Возможны случаи, когда множество операто-ов *G* содержит всего один оператор *g*. Естественно, он является ючевым. Разбиение в этом случае начинается с удачного выбо-а промежуточного состояния *f* и далее по тому же алгоритму.

Итак, для разбиения задачи на подзадачи и построения соответ-твующего *И/ИЛИ* графа нужны ключевые операторы (обычно более ного). Один из способов нахождения операторов, могущих быть ючевыми, состоит в вычислении *различий* между состояниями по ути от $Sн \in S$ к $Sц \in F$. Каждому возможному различию ставится в оответствие оператор (или их множество), который это различие ожет устранить. Цепочка операторов, последовательно устраняю-их различия между *Sн* и *Sц*, называется решением задачи.

3.3. Пример решения задачи

Проиллюстрируем метод разбиения задачи на подзадачи и оиска решений на *И/ИЛИ* графе на примере с обезьяной и бана-ами (см. п.1.5).

Допустим, что исходное описание задачи задается следую-им образом:

$$Sн= (a,b,c,0,0) \in S,$$
$$Sц= (c,c,c,1,1) \in F,$$
$$G=(g1,\ g2,\ g3,\ g4).$$

При этом отметим, что оператор g_1 устраняет различие ме ду состояниями *(a,b,c,0,0)* и *(b,b,c,0,0)*,

g2 - между *(b,b,c,0,0)* и *(c,c,c,0,0)*,

g3 - между *(c,c,c,0,0)* и *(c,c,c,1,0)*,

g4 - между *(c,c,c,1,0)* и *(c,c,c,1,1)*.

Попытаемся далее выделить подзадачи, применяя, напри ер, оператор *g3* (применение операторов может быть произ ольным, поскольку речь идет о механической процедуре). Ему дут соответствовать подзадачи с описаниями:

1) *((a,b,c,0,0), G, (c,c,c,0,0))*, так как оператору *g3* соответст ет состояние *Fg3=(c,c,c,0,0)*.

2) *((c,c,c,0,0),g3, (c,c,c,1,0))*. Здесь *(c,c,c,1,0)=g3(c,c,c,0,0)*. Т есть результат применения ключевого оператора *g3* к состояни *(c,c,c,1,0)*.

3) Остается еще подзадача: *((c,c,c,1,0), G, (c,c,c,1,1))*.

Мы видим, что подзадача 1 не является заключительной (т.е допускает дальнейшее разбиение), а подзадача 3 является элементарной, то есть заключительной, если применить оператор *g4*.

Подзадача 2 также является заключительной.

Разбиваем на подзадачи 1-ю подзадачу. Одно из различий между начальным и целевым состоянием может быть устранен применением, например, оператора *g2* (возможно и *g1*). Получим новые подзадачи с описаниями:

1.1. *((a,b,c,0,0), G, (b,b,c,0,0));*
1.2. *((b,b,c,0,0), g2, (c,c,c,0,0)).*

Здесь мы видим, что подзадача 1.2. является элементарной а различие в подзадаче 1.1. может быть устранено, если приме нить оператор $g_1 \in G$, и тогда она является тоже элементарной.

Целиком граф И/ИЛИ приведен на рис. 3.6.

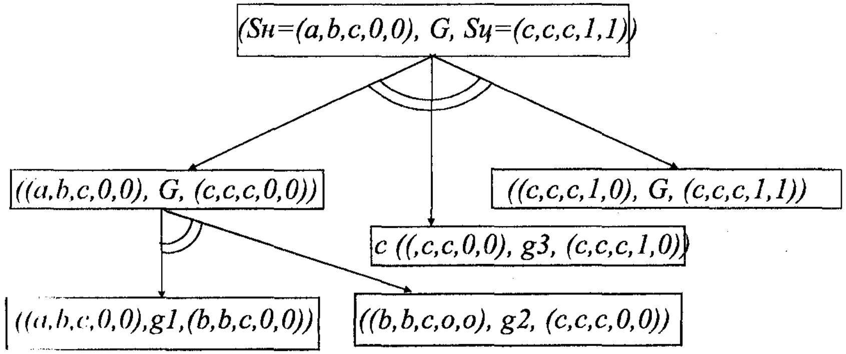


Рис. 3.6. И/ИЛИ граф для задачи с обезьяной и бананами (ключевой оператор - g3)

Точно так же можно было бы осуществить разбиение ис ходной задачи на подзадачи, применяя вначале любой другой оператор *g4*, *g2*, или *g1*. На рис.3.7 показан граф для ключевого оператора *g4*.

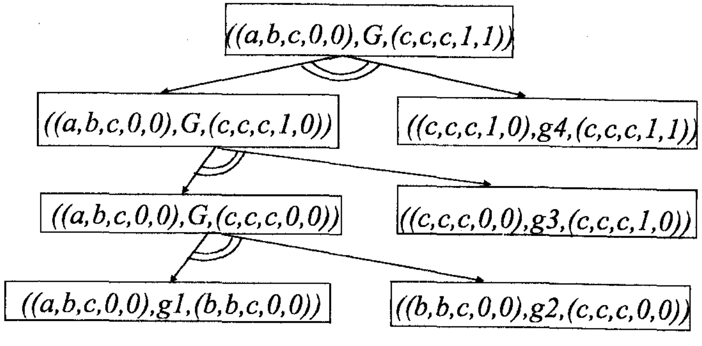


Рис.3.7. И/ИЛИ граф для оператора g4

3.4. Достоинства и недостатки методов поиска в пространстве состояний

Приведем примеры задач, для решения которых могут применяться методы поиска в пространстве состояний.

1) *Комбинаторные задачи*. Классический пример – задача о коммивояжере (п. 2.4.). Состояния задаются списком городов, операторы соответствуют действию: направиться в город ***A***, направиться в город ***B***, ..., направиться в город ***N***. Критерий достижения цели: любое описание, начинающееся и оканчивающееся городом ***A*** и перечисляющее все другие города, есть описание состояния, удовлетворяющего поставленной цели. Данная задача допускает поиск оптимального решения, поскольку дугам могут быть приписаны стоимости (длина пути) перехода из одного города в другой. Данная задача хорошо представима в графической форме (карте расстояний), что и делает ее удобной для применения методов поиска в пространстве состояний. Считается, что метод поиска в пространстве состояний применим при числе городов до 50. К данному классу задач хорошо сводятся практически любые однокритериальные задачи лабиринтного типа.

2) *Задачи синтаксического анализа*. При работе с языками (естественными или искусственными) задается грамматика построения правильных выражений (строк символов – слов, предложений, выражений) в данном языке и, следовательно, способ определения принадлежности произвольной строки символов к этому языку. То-

гда множество состояний задачи синтаксического анализа мож
быть задано как множество строк (слов). Начальное и целевое
стояния определяются какими-то фиксированными словами в ал
вите этого языка. Операторы могут быть заданы в виде правил пе
писывания типа: μ ß æ → μ ø æ, где подстрока ß может быть заме
на подстрокой ø. Эти правила могут выражать, например, синтакс
рассматриваемого языка. Критерий цели для этой задачи мож
быть задан строкой, число символов которой совпадает с числ
символов целевого слова. Пример задачи такого типа: как сло
«море» преобразовать в слово «рыба» путем замены на каждом ша
одной буквы. Очевидно, что при большом числе правил перепис
вания граф пространства состояний становится слишком больши

3) *Задачи распределения*. Здесь известно: объем продукц
каждого типа; величина поставок, которые должны быть осущес
лены в заданные пункты. Необходимо найти такое распределен
поставок, при котором, например, затраты на перевозки, были
минимальны. Состояния описываются списком величин избыт
ной продукции, которая имеется в заданных пунктах. Операто
соответствуют передаче избытка продукции из одного пункта
другой. В качестве критерия может быть взято целевое состоян
при котором будут удовлетворены все заявки на поставку треб
мой продукции. Типичная задача линейного программирования.

4) *Задачи управления* типа: требуется перевести объ
управления с начальными значениями установленных параметр
процесса в состояние, при котором эти параметры будут име
заданные значения. Задача об обезьяне и бананах является т
пичной задачей этого класса.

Возникает вопрос, когда же следует применять описанн
здесь методы поиска в пространстве состояний? Для каких клас
задач эти методы применимы и могут дать эффективные решения

Во-первых, это должны быть задачи, для которых так
представление возможно и является естественным. Это означа
что для выбранной задачи необходимо найти способ описан
любых потенциально возможных состояний предметной облас
определить множество операторов (ключевых операторов для м
ки разбиения задачи на подзадачи), с помощью которых

редметная область может переходить из одного состояния в
ругое, определить критерий достижения цели.

Во-вторых, для этих задач не существуют или неизвестны
етоды, которые были бы более эффективны. Таким образом, мы
пускаем, что одна и та же задача может решаться и другими
етодами, но по каким-то своим особенностям или соображени-
эти методы представляются нам лучшими.

◆ Если найден удачный способ представления задачи, важно
же, чтобы пространство состояний было не слишком большим.
уществует множество примеров задач, кажущихся трудными, но
ких, что при правильной их трактовке, соответствующие про-
ранства состояний оказываются очень узкими. Подчас простран-
во состояний «сжимается в точку» после того, как обнаруживает-
ся, что некоторые операторы могут быть выброшены за ненадобно-
стью, а другие – объединены в более мощные операторы. И даже,
ли такие простые преобразования неосуществимы, может ока-
ться, что полная переформулировка задачи, изменяющая само
понятие состояния, приведет к меньшему пространству. Таким об-
ом, проблема поиска хорошего представления имеет важнейшее
ачение. Поиск его всегда опирается на специфику задачи, ис-
пользует ее кардинальным образом. Это означает, что способ пред-
тавления всегда носит уникальный характер, по сути, является
обым искусством и, следовательно, вряд ли существуют какие-то
щие правила поиска хороших представлений.

Во всех рассмотренных нами классах задач, для решения ко-
орых обычно применяются методы поиска в пространстве состоя-
ний, число свойств объектов предметной области и отношений ме-
ду ними, служащих затем основой для распознавания ситуаций,
гда небольшое. Например, в задаче об обезьяне и бананах – это
оординаты (Обезьяны, Ящика, Бананов) и два отношения – «*На*
одиться НА», «*Находиться У*». В задаче коммивояжера использо
лось: свойство «*Иметь имя*», а в качестве отношения – расстоя
ние между каждыми двумя городами. В задаче распределения это
мена пунктов нахождения продукции и поставок, объемы имею
щихся и требуемых поставок. Очевидно, что при увеличении числа
ойств, отношений и их значений, размерность пространства со

стояний будет быстро возрастать таким образом, что применен переборных методов станет практически нереализуемым.

◆ В этой связи в искусственном интеллекте развивались и др гие, более универсальные, модели представления знаний (МПЗ) задаче. Общие требования к МПЗ можно сформулировать сл дующим образом:

1) необходимы способы представления знаний о задач *безразличные к содержанию* (смыслу) самих знаний. Это позв лит применять эти способы для представления знаний в любь предметных областях для решения задач;

2) эти способы должны иметь *механизм логического анал за*, позволяющий моделировать человеческую логику выработк решений на выбранных моделях представления знаний;

3) эти способы должны решать задачи, которые известным формальными моделями и методами не решаются в связи с их н строгостью и нечеткостью.

Первое требование означает, что проблема смысла (содерж ния) знаний заменяется проблемой синтаксического их представл ния. Второе требование предполагает возможность разделения с мих знаний и механизмов их логического анализа. Третье требов ние предполагает возможность формализации опытного (экспертн го) знания, накапливаемого специалистами различных предметны областей и имеющих качественный (а не количественный) характе Именно с таким знанием не могут работать все известные формал ные математические модели. Мы, таким образом, приходим к нео ходимости разработки и применения *неформальных моделей знани*

Вопросы для самопроверки и упражнения

1. В каких случаях рекомендуется применять метод разби ния на подзадачи?
2. Дайте определение И-вершине, ИЛИ-вершине.
3. Какова структура И/ИЛИ графа?
4. Какая вершина называется заключительной?
5. Каково условие раскрытия И-вершины?
6. Каково условие раскрытия ИЛИ-вершины?
7. Какая вершина называется разрешимой?

8. Какой оператор является ключевым?
9. Объясните идею разбиения на подзадачи на линейном графике.
10. Постройте И/ИЛИ граф для задачи с обезьяной, начиная оператора g2.
11. Задача о выборе маршрута.

Известно, что применение метода сведения задачи к подзадачам особенно эффективно, когда подзадачи взаимно независимы. Покажем это на примере отыскания маршрута между городами *A* и *T* (см. рис. 3.8).

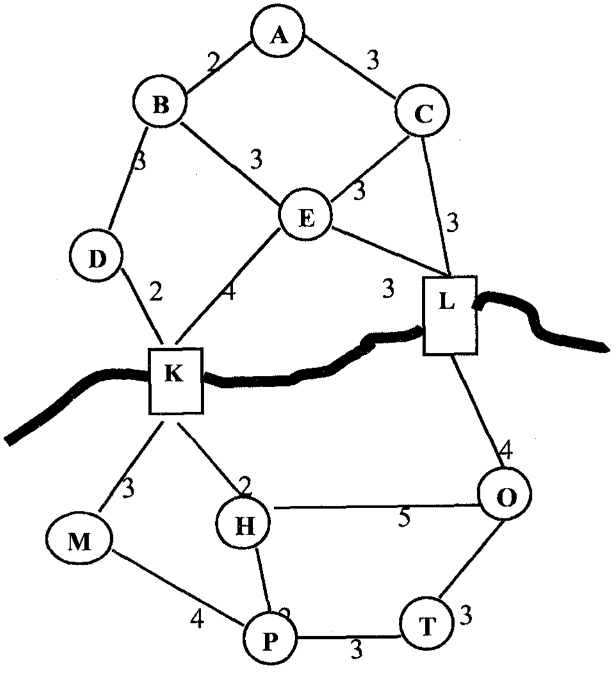


Рис. 3.8. Задача о выборе маршрута

На рисунке задана маршрутная сеть между населенными

пунктами, лежащими на двух берегах реки. K и L – это мосты, с
зывающие города одного берега с другим. На дугах между пункта
указаны расстояния (стоимость). При построении дерева подзад
следует учитывать, что добраться до из пункта A в пункт T мож
двумя альтернативными путями: через K и через L. Постройте гр
И/ИЛИ для данной задачи, определите кратчайшие пути.

12. Вернемся к задаче о квасе (гл. 2). Требуется раздели
квас, находящийся в полном бочонке объемом 8 литров, попола
Для этого имеются два пустых бочонка 5 и 3 л. Попробуйте р
шить эту задачу на этот раз методом сведения к подзадаче.

13. Постройте граф И/ИЛИ для задачи о коммивояжере
2.4), начиная разбиение, к примеру, с пункта ***E***.

4. МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

4 .1. Знания как объект исследования и преобразования

В определении модели предметной области мы использова
понятие "знание", никак не определив его и не указав, чем отлича
ся "знание" от каких-либо других понятий, таких как информац
данные, сообщение, суждение и т.п.. Посмотрим, что об этом го
рит "Философский словарь" (под ред. Фролова И.Т. М. 1986 г.).

"Знание - идеальное выражение в знаковой форме объект
ных свойств и связей мира, природного и человеческого. П
этом знания могут быть донаучными (житейскими или опыт
ми) и научными, которые, в свою очередь, делятся на эмпири
ские и теоретические".

В нашем понимании знания определяются более конкрет
*знанием является информация, которая отражает объ
тивные* ***свойства*** *и* ***связи*** *некоторых объектов, явлений, п
цессов, сущностей и отношениями между ними как в субъект
ном, так и научном (объективном) выражении.*

Нас будут интересовать также *системы знаний* о предмет
области (объектах, связях между ними, структуре и закономерно
поведения, взаимодействии со средой и т.п.), обеспечивающих ре
ние целевых задач. Под *системой знаний* понимается совокупн
знаний, образующих целостное описание некоторой проблемы с

ной и достаточной степенью точности [2]. Целостность означает,
между отдельными сведениями, входящими в знание, существуют
и, следовательно, одни сведения могут быть выводимы из дру-
Это также означает, что система знаний есть информационная
тема (информационная модель), отвечающая требованиям полноты
непротиворечивости. В случае, если между отдельными сведениями
связи не установлены, то они превращаются в единичные фак-
некоторые суждения, относительно которых можно судить об их
инности или ложности. Такие суждения можно назвать данными.
ким образом, данное есть вырожденный случай знания. Например,
между деталями автомобиля не указаны связи, то мы получим
росто Базу Данных о деталях автомобиля. Связность между элемен-
ми удобно отображать с помощью ориентированных графов. И то-
результирующий связный граф можно рассматривать как модель
ния о чем-то. Очевидно, что невозможно построить модель всего
ния и мы можем говорить лишь о предметно - или проблемно -
нтированном фрагменте знания для целевого применения. Такие
менты мы будем называть моделями предметных областей.

В ЭВМ *знания*, так же, как и *данные*, отображаются в знако-
форме - в виде формул, текста, файлов, информационных
сивов и т.п.. Поэтому можно сказать, что *знания - это особым
ом организованные данные.*

Но это было бы слишком узкое понимание. А между тем, в
темах искусственного интеллекта знания являются основным
ктом формирования, обработки и исследования. *База знаний,*
ение с базой данных, - необходимая составляющая программ-
комплекса ИИ. Машины, реализующие алгоритмы ИИ, на-
ют *машинами, основанными на знаниях*, а подраздел теории
ИИ, связанный с построением экспертных систем, - *инженерией*
ний. Так чем же отличаются знания от данных?

Обычно под *данными* понимают информацию, определенным
ом подготовленную и представляющую собой объект, отлич-
от команд. Сюда относится все, что обрабатывается программой
ранды, файлы и т.п.) по заданному алгоритму. Этот алгоритм
ет быть очень сложным, существуют даже специальные системы
ения базами данных (СУБД). Но эти системы не могут делать.

одного, с точки зрения ИИ, самого главного. Они не могут модели вать рассуждения, т.е., иначе говоря, проводить *логический выв* Потребовалось достаточно большое время, прежде чем *данные,* степенно эволюционируя, превратились в *знания.* При этом они п обрели, как минимум, шесть обязательных свойств:

1) внутреннюю интерпретацию,
2) внутреннюю структуру связей,
3) внешнюю структуру связей,
4) шкалирование,
5) погружение в пространство с «семантической метрикой»,
6) наличие активности.

В приведенном списке легко угадывается стремление как мож полнее отобразить не только интересующие нас объекты исследо ния, но и самые разнообразные связи и отношения между ними. Т в задаче о наполнении ведра, например, мы имели дело не тольк объектами - ведром, раковиной, краном, - но и отражали отношен между ними *(РЯДОМ, В, НА)*, их состояния *(ПУСТО, ПОЛН ОТКРЫТ, ЗАКРЫТ)*. Будучи описанной на соответствующем язы та задачка уже представляла бы некоторое подобие базы знаний. же можно сказать о любой модели предметной области вообще.

Рассмотрим указанные свойства базы знаний подробнее.

1. *Внутренняя интерпретация*

Это понятие легко пояснить на следующем примере, котор представляет собой, в сущности, обычную таблицу (см. рис. 4.1

Тип прибора	Характеристики		
	Проводимость	Максим. ток, *А*	Максим. напряжение, *В*
КТ837А	p-n-p	7,5	80
КТ817Г	n-p-n	3	25
КП308В	n-тип	0.02	25
КП304А	p-тип	0.03	20

Рис. 4.1. Пример внутренней интерпретации данных

Каждую строчку данной таблицы, так же, как и столбец, будем рактовать как отдельную информационную единицу (ИЕ), расположенную в памяти машины. Тип транзистора, указанный в начале троки, является именем ИЕ-строки, имя ИЕ-столбца указано сверху.

〈Имя *m*-го слота, Значение *m*-го слота〉).

Информационная единица для первой строки с именем Т837А будет выглядеть так: (КТ837А 〈Проводимость, p-n- 〈Макс. ток, 7.5 *А*〉 〈Макс. Напряжение, 80 *В*〉), для столбца с менем Проводимость соответственно: (Проводимость 〈КТ837А, n-p〉 〈КТ817Г, n-p-n〉 〈КП308В, n-тип〉 〈КП304А, p-тип 〉). руглые скобки, таким образом, выделяют содержание ИЕ, а уговые скобки - определенные самостоятельные части, называеные *слотами*. В общем виде полная запись ИЕ будет выглядеть ак: (Имя ИЕ 〈Имя первого слота, Значение первого слота〉 Имя второго слота, Значение второго слота〉…).

База данных, структурированная таким образом, еще не база наний, но уже может ответить на некоторые вопросы, например, дать информацию о характеристике транзистора или подобрать транзисторы по их проводимости. Другими словами, она уже ожет отвечать на вопросы, касающиеся содержимого ее памяти.

2. *Наличие внутренней структуры связей*

Представим теперь, что в качестве слотов у нас выступают ругие информационные единицы. В этом случае слоты будут ы вкладываться друг в друга, как в «матрешке». Полученная руктура получила название *фрейма* (см. рис. 4.2).

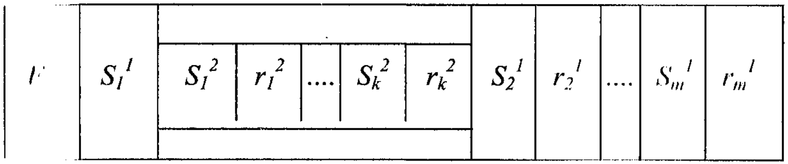


Рис. 4.2. Структура фрейма

На рисунке: *F*- имя фрейма, S_1^1 - имя первого слота, Si^2 - имя второго слота, который в свою очередь содержит *k* слотов второго уровня вложенности.

Пользуясь предыдущим примером, нетрудно представить себ фрейм, скажем, с четырьмя уровнями вложений. 1-й уровень «видовое» название «Транзисторы». 2-й могли бы составить д слота - «Биполярные» и «Униполярные». 3-й уровень вложения слоты с именами «КТ837А», «КТ817Г», «КП308В», «КП304А», в качестве 4-го уровня - слоты, отражающие характеристик транзисторов согласно рис. 4.1.

Между слотами различных уровней могут быть самые разны отношения. Вложенным может быть не только слот, но и фрей имеющий уже свои многоуровневые вложения. Наконец, вложе ной может быть команда или даже целая программа. Все это пр дает фреймовым структурам большую гибкость и многосвязность.

3. Наличие внешней структуры связей

Из предыдущего следует, что при работе с фреймами могут во никнуть такие ситуации, когда отдельные факты и явления, входящ в структуру одного фрейма, вступают в ситуационную связь с факт ми и явлениями, описанными в структуре другого фрейма. Для от бражения таких связей используются отдельные слоты. В них указ ваются имена фреймов, с которыми есть связь, и имена отношени осуществляющих их. Так возникает сеть с именами фреймов в ве шинах. С помощью дуг, над которыми написаны имена соответс вующих отношений, вершины соединяются между собой, образ так называемую *семантическую сеть*. Попробуем ее построить. Как там у Пушкина?

> Вот едет могучий Олег со двора,
> С ним Игорь и старые гости,
> И видят: на холме, у брега Днепра,
> Лежат благородные кости;
> Их моют дожди, засыпает их пыль,
> И ветер волнует над ними ковыль.

На рисунке 4.3 обозначены следующие понятия (если хотит фреймы): СтГ - старые гости, Дв - двор, Кос - кости (благоро ные), Пл - пыль, Дж - дождь, Кв - ковыль, Вт - ветер, Хл - хол Бг - берег Днепра.

Множество G описывает систему отношений: *g1 - быть вм*

те (двойная дужка объединяет обе части этого соотношения), *g2 - выезжать со, g3 - видеть, g4 - быть на, g5 - быть у, g6 - засыпать, g7 - мыть, g8 - волновать, g9 - быть над.*

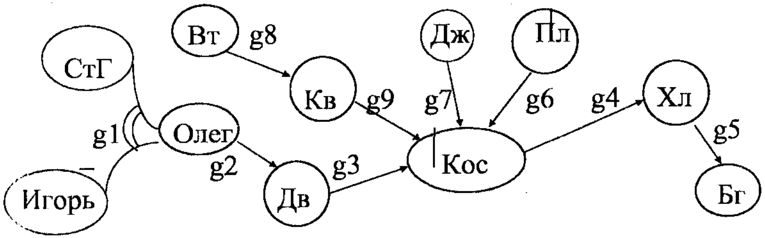


Рис.4.3. Пример семантической сети

«Семантический» значит «с учетом смысла слов». Указанная сеть так и построена. При серьезных задачах она принимает весьма «запутанный» вид, связи ее усложняются. Построенные нами ранее графы решения различных задач (см. гл. 2) вполне могли бы послужить основой для построения элементарных семантических сетей.

Обратим внимание на тип отношений G между понятиями (фреймами) одной сети. Они могут быть самыми разными. Если, например, они отражают причинно-следственные связи, то такие сети называются *сценариями*. Если эти отношения отражают связи по включению (типа «принадлежать к классу», «состоять из»), то это будут *иерархические сети*. Отношения могут быть еще и такими, которые связывают аргументы и значения функции. Такие сети называются *вычислительными моделями*.

4. Шкалирование

Разумной деятельности человека свойственно стремление к упорядоченности. Мы пытаемся как-то систематизировать, «разложить по полочкам» те явления, события, факты, а точнее - информационные единицы, с которыми мы имеем дело. Для этой цели мы используем разного рода шкалы. Это могут быть строгие *метрические шкалы,* такие как шкала упорядочения людей по возрасту, шкала воинских званий или, к примеру, шкала весовых категорий в боксе (наилегчайший вес, легчайший, полулегкий и т.д.). Но это могут быть и «размытые шкалы», такие естественные в нашем языке. Но

как мы оцениваем частоту появления какого-то события: *Никогд*
Чрезвычайно редко, Очень редко, Редко, Редко, но не очень, Не ча
то - не редко, Часто, но не слишком, Часто, Очень часто, Почт
всегда, Всегда. А еще есть так называемые *«оппозиционные шкал*
типа *Хороший - Плохой, Острый - Тупой, Сильный - Слабый* и т.
Сюда же относится и знакомое всем нам *Мы - Они.*

Отображая отношения и связи реального мира, база знани
должна уметь отразить и это свойство человеческого мышления

5. Погружение в пространство с «семантической метрикой

В нашем сознании это пространство образуют понятия, факт
явления, близкие по своему смыслу (семантике). Честно говор
«метрика» эта не слишком строгая и даже в чем-то противоречивая

Возьмем понятие *учитель*. В какое пространство (чаще гов
рят - *кластер*) мы его поместим? Это может быть «Интеллиге
ция»,или «Работники умственного труда», или «Образованные лю
ди», или просто «Служащие». Конечно, все зависит от конкретн
типовой ситуации, но мы все-таки отметим этот факт: точки ка
дого кластера образуют совокупности понятий, семантически бли
ких между собой. Таково свойство нашего мышления. *Типовая си*
туация - ядро, вокруг которого группируется информация.

Но есть и другой принцип образования кластеров. Выбор то
или иного понятия из множества ему близких подчиняется у челов
ка еще и частоте появления. На просьбу назвать поэта почти кажды
ответит: Пушкин; на просьбу назвать фрукт - чаще всего - яблоко.

Таким образом, имеются, по крайней мере, две систем
оценки близости информационных единиц. Одна опирается на
ситуативную близость, а другая - на частоту появления тех и
иных понятий в типовых ситуациях.

Разработана довольно стройная теория моделирования обр
зования ситуационных кластеров. Мы не будем сейчас вдаваться
подробности. Отметим лишь то, что метод погружения в простра
ство с семантической метрикой, будучи, хотя и приближенно, ре
лизованным в базе знаний, существенно повышает ее «интеллект
альный уровень».

6. Наличие активности

Как работает «обычная» ЭВМ? Очень просто - по заданно

ограмме она обрабатывает те или иные *данные*. В программе
средоточено *процедурное знание*. Оно хранит информацию о
м, *как* надо действовать, чтобы получить желаемый результат.
нные представляют собой *декларативные знания*. Они хранят
формацию о том, *над чем* надо выполнять эти действия. Про-
амма, таким образом, играет роль активатора данных.

Иное дело у человека. Очень часто в процессе мышления
кларативные знания у него являются активатором процедур-
ых. Реализовать подобную возможность в базе знаний, как гово-
тся, в чистом виде пока не удается. Используются *смешанные*
едставления, в которых декларативные и процедурные знания
нимаются единообразно и могут активизировать друг друга.

В качестве примера рассмотрим фрейм, содержащий несколь-
слотов. Допустим теперь, что в качестве одного из слотов вы-
упает имя какой-либо процедуры, подлежащей исполнению. Но,
к мы видели на примере семантической сети, обращение к тому
ному слоту (или фрейму) определяется множеством разнооб-
ных отношений как между слотами одного фрейма, так и между
угими фреймами. Все это как раз и формирует условия, необхо-
имые для выполнения указанной процедуры. Иными словами,
кларативные знания становятся активаторами процедур.

Возможность активизации процедур посредством данных отме-
ем как важнейшее свойство базы знаний. В сущности, это означает
ход от фундаментального принципа работы классической ЭВМ
неймановского типа - работы по жестко заданному алгоритму. С
ого момента мы переходим к принципиально новым системам об-
ботки информации - системам искусственного интеллекта.

7. Продукционные системы

Продукционная модель представления знаний основана на
теме правил, называемых продукциями. Как база знаний она
адает всеми известными свойствами, изложенными выше, но
еет своеобразную структуру:

$$P(x,y)\longrightarrow A.$$

Здесь $P(x,y)$ есть логическая функция, A - некоторое сужде-
ие, выражающее рекомендацию лицу, принимающему решени

или решающей системе. То есть, если для некоторых значен переменных x, y, например, $x = a$, $y = b$, логическая функц $P(a,b)$ принимает значение «истина», то справедливо некотор суждение (совет) A. Иначе продукцию можно истолковывать к суждение типа «если..., то...».

Левая часть продукции называется прототипом или образц состояния, с которым сопоставляются реальные состояния. Пра часть называется рекомендацией (или решением), которое след принять в случае, если конкретное состояние предметной обла сопоставляется с образцом (интерпретируется). Совокупность ких вот правил - продукций и образует базу знаний (БЗ).

Основной механизм логического вывода - это механизм соп тавления конкретных состояний предметной области с образцами При этом возможны две основных стратегии вывода: «снизу-вве (от ситуации → к цели) и «сверху-вниз» (от цели ⟶ к ситуации)

Бабушка говорит «Красной Шапочке» (Волку): «Дерни веревочку - дверь и откроется». Для нас это продукция: «Е дернуть за веревочку (*p*), **то** дверь откроется (*q*)»: $p \to q$. И возник вопрос: что будет, «если *дернуть за веревочку*»? От определяется путем сопоставления с левой частью продук (образцом) *p*. Совпало. Ответ: будет «*q - дверь откроется*» (стр тегия «снизу-вверх»). Но может быть другой вопрос: «В ка случае *дверь откроется*?». Ответ находтся путем сопоставле с правой частью продукции. Совпало. Ответ: «*p - дернуть за* *ревочку*» (стратегия «сверху-вниз»).

Правила-продукции не зависят друг от друга, и поэтому кая БЗ легко пополняется и модифицируется. Подробно прод ционные системы будут рассмотрены в другом пособии.

4.2. Классификация моделей представления знаний

Как организовать базу знаний с тем, чтобы она соответст вала требованиям СИИ, мы приблизительно познакомились. (видно, что методы представления знаний в базе должны соот ствовать изложенным требованиям.

Существуют три типа моделей представления знаний (МП формальные модели представления знаний,

- неформальные (семантические, реляционные) МПЗ,
- интегрированные МПЗ.

К формальным МПЗ относятся модели, построенные на основе *исчисления высказываний* и *исчисления предикатов*.

К неформальным (реляционным, семантическим) относятся:

- продукционные модели,
- семантические сети,
- фреймовые МПЗ.

Интегрированные МПЗ совмещают в себе модели различных типов.

4.2.1. Неформальные модели представления знаний

Все методы представления знаний, которые мы рассматривали выше, включая продукции, относится к неформальным моделям. В отличие от формальных моделей, в основе которых лежит строгая математическая теория, неформальные модели такой теории не придерживаются. Каждая неформальная модель годится только для конкретной предметной области и поэтому не обладает той универсальностью, которая присуща моделям формальным. Логический вывод – основная операция в СИИ – в формальных системах строг и эффективен, поскольку подчинен жестким аксиоматическим правилам. Вывод в неформальных системах во многом определяется самим исследователем, который и отвечает за его корректность.

Каждому из методов ПЗ соответствует свой *язык представления знаний* (ЯПЗ). Их четыре типа: *логические, сетевые, фреймовые, продукционные*. Рассмотрение сетевых, фреймовых и продукционных ЯПЗ выходит за рамки данной книги. Достаточно отметить, что все они, в конечном итоге, приходят к использованию языков формальной логики, что подчеркивает важность логических ЯПЗ в решении интеллектуальных задач.

4.2.2. Формальные модели представления знаний

Система ИИ в определенном смысле моделирует интеллектуальную деятельность человека и, в частности, - логику его рассуждений. В грубо упрощенной форме наши логические построения при этом сводятся к следующей схеме: из одной или нескольких посылок (которые считаются истинными) следует сделать "логически

верное" заключение (вывод, следствие). Очевидно, для этого нео
ходимо, чтобы и посылки, и заключение были представлены на п
нятном языке, адекватно отражающем предметную область, в ко
рой проводится вывод. В обычной жизни - это наш естественны
язык общения, в математике, например - это язык определенн
формул и т.п. Наличие же языка предполагает, во-первых, налич
алфавита (словаря), отображающего в символьной форме весь наб
базовых понятий (элементов), с которыми придется иметь дело,
во-вторых, набор синтаксических правил, на основе которых, пол
зуясь алфавитом, можно построить определенные выражения.

Логические выражения, построенные в данном языке, мог
быть истинными или ложными. Некоторые из этих выражени
являющиеся всегда истинными, объявляются *аксиомами* (или *п*
стулатами). Они составляют ту базовую систему посылок, исх
дя из которой и пользуясь определенными правилами вывод
можно получить заключения в виде новых выражений, также я
ляющихся истинными.

Если перечисленные условия выполняются, то говорят, ч
система удовлетворяет требованиям *формальной теории*. Ее т
и называют *формальной системой* (ФС).. Система, построенн
на основе формальной теории, называется также *аксиоматич*
ской системой.

Формальная теория должна, таким образом, удовлетворя
следующему определению:

всякая формальная теория $F = (A,V,W,R)$, определяющая н
которую аксиоматическую систему, характеризуется:

наличием алфавита (словаря), A,

множеством синтаксических правил, V,

множеством аксиом, лежащих в основе теории, W,

множеством правил вывода, R.

Исчисление высказываний (ИВ) и исчисление предикат
(ИП) являются классическими примерами аксиоматических си
тем. Эти ФС хорошо исследованы и имеют прекрасно разраб
танные модели логического вывода - главной метапроцедуры
интеллектуальных системах. Поэтому все, что может и гарант
рует каждая из этих систем, гарантируется и для прикладных Ф

моделей конкретных предметных областей. В частности, это гарантии непротиворечивости вывода, алгоритмической разрешимости (для исчисления высказываний) и полуразрешимости для исчисления предикатов 1-го порядка).

ФС имеют и недостатки, которые заставляют искать иные формы представления. Главный недостаток - это «закрытость» т.е. их негибкость. Модификация и расширение здесь всегда связаны с перестройкой всей ФС, что для практических систем сложно и трудоемко. В них очень сложно учитывать происходящие изменения. Поэтому ФС как модели представления знаний используются в тех предметных областях, которые хорошо локализуются и мало зависят от внешних факторов.

В заключение скажем, что выбор модели представления знаний при построении СИИ является ответственейшим этапом. Эта работа выполняется специалистом по профессии «инженер по знаниям» - новая специальность, появившаяся в результате развития СИИ и экспертных систем. Вопросы технологии проектирования СИИ будут рассмотрены в другом издании.

Вопросы для самопроверки и упражнения

1. Дайте определение понятию «знание».
2. Чем принципиально отличаются «знания» от «данных»?
3. Какими основными свойствами должны обладать БЗ?
4. Приведите пример внутренней интерпретации знаний.
5. Как записывается простейшая информационная единица?
6. Что такое «слот»?
7. Какова структура фрейма?
8. Как в БЗ отображается внешняя структура связей?
9. Почему сеть называется семантической?
10. Приведите пример семантической сети
11. В чем заключается шкалирование данных?
12. Как вы понимаете «пространство с семантической метрикой»?
13. Что такой кластер?
14. Что содержат процедурные знания?
15. Что содержат декларативные знания?

16. В чем заключается принцип активности?
17. Что представляет собой продукционная модель?
18. Как организован вывод на продукциях?
19. Перечислите методы представления знаний.
20. Дайте определение формальной системе.
21. Какими свойствами характеризуется ФС?
22. Попробуйте составить семантическую сеть для следу щих фраз:

а) И веют древними поверьями
Ее упругие шелка,
И шляпа с траурными перьями,
И в кольцах узкая рука.

б) Транзистор – электронный прибор, имеющий тр более электрода и выполняющий операции усиления, ге рирования и преобразования электрических колебаний.

в) Идет новобрачный к теще, а у нее уже новобрач сидит разнаряжена, и фрукты стоят, и теща вино подноси зятю, и тысяцкому, и поежанам.

г) Московиты настаивают на том, чтобы все крепости, тые Вашим Величеством, были возвращены их Великому княз

5. ИСЧИСЛЕНИЕ ВЫСКАЗЫВАНИЙ КАК МЕТОД ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

5.1. Определение высказываний

Под высказыванием обычно понимают некоторое сооб ние, предложение, выражение, утверждение и т.п. Но такое рокое толкование нам не подходит, его надо конкретизиров Если, услышав сообщение *«идет дождь»*, вы просто приняли к сведению и перед выходом из дома взяли зонт, то с точки ния ИВ это сообщение не является высказыванием. Если же сле указанных слов вы подошли к окну, чтобы удостоверитьс их истинности, то такое сообщение уже можно определить им но как высказывание. Итак, всякое высказывание предпола оценку его истинности (ложности).

Простые, односложные высказывания обозначаются стр

ыми латинскими буквами *a*, *b*, *m*, *l*, *r*, *q*... Это высказывания ти- а *«светает»*, *«дом белый»*, *«сегодня холодно»*, *«Петя любит ицу»*. Если установлено, что высказывание *r* истинное, то пи- шут $r = И$ (или $r = 1$), если *r* ложное, то $r = Л$ ($r = 0$).

Каждое высказывание может быть либо истинным, либо жным, третьего не дано - таков закон "исключенного ретьего".

Простые высказывания называют еще элементарными или омарными.

Дадим такое определение:

ксиоматическая система, предназначенная для моделирова- ия и изучения логики высказываний, называется *исчисле- ием высказываний.*

Начнем теперь определять формальную теорию, лежащую в нове исчислении высказываний.

5.2. Алфавит исчисления высказываний

Алфавит *А* часто называют словарем. Этот словарь, во- ервых, содержит сами высказывания *a*, *b*, *m*, *n*,... Во-вторых, он держит еще так называемые логические связки.

ни хорошо известны читателю по курсам дискретной математи- и логических основ ЭВМ, программирования и др. Мы все-таки речислим их, тем более что в ИИ они используются в расши нном составе и в разных обозначениях (см. таблицу 1).

Таблица 1

Название	Обозначение	Тип
Отрицание	‾, ¬, *not*, *НЕ*	унарный
Конъюнкция	∧, &, *and*, *И*	бинарный
Дизъюнкция	∨, +, *or*, *ИЛИ*	бинарный
Импликация	→	бинарный
квивалентность	≡	бинарный

Тип отрицания унарный, потому что оно может быть при ожено к одному высказыванию, даже если это высказывание ожное выражение. Остальные - бинарные, т.к. они могут со инять как минимум два высказывания. Действия связок опре ляются таблицей истинности.

Таблиц

x	y	$x\vee y$	$x\wedge y$	$x\to y$	$y\equiv x$	$\overline{x}\vee y$
И	*И*	*И*	*И*	*И*	*И*	*И*
И	*Л*	*И*	*Л*	*Л*	*Л*	*Л*
Л	*И*	*И*	*Л*	*И*	*Л*	*И*
Л	*Л*	*Л*	*Л*	*И*	*И*	*И*

В данном пособии мы будем употреблять следующие о значения

отрицания: $\overline{X}$;
конъюнкции: $X\wedge Y$ или XY, или $X\cdot Y$;
дизъюнкции: $X\vee Y$ или $X+Y$;
импликации: $X\to Y$;
эквивалентности: $X\equiv Y$.

С помощью связок из простых высказываний можно строить б лее сложные высказывания, употребляя при этом еще и разл ные скобки:

$$(q\to p)\to[(p\wedge q)\to r];$$
$$\{[(a\vee b)\wedge m]\to b\}\to a.$$

Не будем останавливаться на хорошо всем известных тол ваниях связок конъюнкции (*И*) и дизъюнкции (*ИЛИ*). Замет только, что за исключением импликации все остальные свя подчиняются перестановочному закону, т.е. не меняют сво значения, если x и y поменять местами.

Особый разговор об импликации. Условно она моделиру утверждение типа "если x..., то y...". Условность здесь заключа ся в том, что не всегда поведение её очевидно. Например, вы жение: "если x - *ЛОЖЬ*, то y -*ИСТИНА*" оказывае *ИСТИННЫМ*. Тем не менее, эта связка широко применяется, п сколько отражает одно из основных логических понятий.

Импликация неперестановочна. Пусть мы имеем такое в сказывание: "если два слагаемых нечетные (r), то их сумма - ч ная (q)". Оно представляется импликацией $r\to q$, которая, о видно, истинна: $(r\to q)=И$. Попробуем поменять местами r и $q\to r$. Новая импликация означает: "если сумма двух слагаем четная (q), то оба слагаемых нечетные (r)", что верно только о

ности. Импликация отображает причинно-следственные связи и предостерегает нас от широко распространенной логической ошибки, когда причину и следствие меняют местами. Первую часть импликации называют посылкой или *антецедентом*, вторую - заключением или *консеквентом*.

Обратимся к приведенной выше таблице истинности и отметим следующее:

- если посылка импликации истинна, то значение истинности импликации совпадает со значением заключения;
- если посылка ложна, импликация всегда истинна;
- импликация ложна только при одном значении посылки (*И*) и заключения (*Л*);
- если посылка истинна и вся импликация истинна, то и заключение истинно (правило заключения или *modus ponens*).

Кстати говоря, импликацию, так же, как и эквивалентность, можно исключить:

$$(x\to y)=(\overline{x}+y);\quad (r\equiv p)=(r\to p)(p\to r)=(\overline{r}+p)(\overline{p}+r).$$

Первое равенство следует из сравнения соответствующих колонок таблицы 2, второе предлагаем доказать самостоятельно.

5.3. Синтаксис исчисления высказываний

Речь идет о правилах V построения сложных высказываний. Как уже говорилось, такие высказывания строятся (не считая скобок) с помощью логических связок - и только. Естественно назвать такие сложные высказывания формулами. Но в ИВ в целях общности рассуждений формулами называют также и атомарные высказывания типа r, p и т.п. Если речь идет вообще о формулах, то их обозначают через прописные латинские буквы: A, B, N, P...

Из сказанного можно сформулировать следующие синтаксические правила:

(1) *всякое высказывание есть формула;*

(2) *формулы, построенные с помощью логических связок и скобок, называются* ***правильно построенными формулами (ппф)***.

Второе правило вводится единственно для того, чтобы подчеркнуть необходимость построения новых формул только с по мощью известных пяти связок. Стараясь подчеркнуть униве сальность этих связок, их "формулообразующие" свойства, э связки часто называют *пропозициональными*.

Два указанных правила определяют синтаксис языка исчисл ния высказываний. В любых языках синтаксис дает возможнос распознавать фразы среди различных наборов слов. В нашем случ он определяет формулы, которые здесь являются аналогами фраз.

Но нас интересует еще и семантика языка, т.е. определен значений, которые принимают формулы. Поскольку каждое эл ментарное высказывание может принимать два значения: либо либо *Л*, то формулы, построенные на их основе и являющиеся пп очевидно, также будут иметь только два значения: *И* или *Л*: знач ние сложной формулы есть функция значений ее составляющих.

Рассмотрим, например, ппф импликации $p \to q$. При знач ниях $p = И$ и $q = Л$ эта формула принимает значение *Л*, при вс других - значение *И*. Приписать p и q какие-то определенные зн чения *И* (*Л*) означает задать *интерпретацию* формулы.

Скажем, дизъюнкция $a + b$ принимает значение *Л* только п интерпретации $a = Л$ и $b = Л$.

Если некоторая формула принимает значение *И* при любо интерпретации входящих в нее атомарных высказываний (фо мул), то она называется *общезначимой*, *универсально-истинно тождественной* или *тавтологией*. Примеры тавтологий:

$$(a + a) \to a;\ (a + b) \to (b + a);\ (a + \bar{a}).$$

Если формула принимает значение истины хотя бы при о ной интерпретации, она называется *выполнимой*. Дизъюнкция конъюнкция - выполнимые формулы. Все элементарные выск зывания выполнимы по определению.

Есть, однако, формулы, которые ложны при всех интерпр тациях, например, конъюнкция $x \wedge \bar{x}$. Такие формулы называю ся *невыполнимыми* или *противоречивыми*. Отрицание общезн чимой формулы - невыполнимая формула.

5.4. Преобразование формул

Успешное решение проблем логического вывода требует пре ставления формул в определенном виде, для чего их необходим

реобразовывать, естественно, сохраняя истинность. Правила реобразования логических формул широко известны под назва- ием булевой алгебры. Здесь мы просто перечислим их, иногда пуская знак конъюнкции, как это принято при обозначении ум- ожения в алгебре, а дизъюнкцию обозначая через «+».

Для дизъюнкции имеем равенства:

$Л = Л$, $x + Л = x$,
$И = И$, $x + И = И$,
$Л = И$, $x + x = x$,
$И = И$, $x + \bar{x} = И$.

Для конъюнкции соответственно:

$Л = Л$, $x \wedge Л = Л$,
$И = Л$, $x \wedge И = x$,
$Л = Л$, $x \wedge x = x$
$И = И$, $x \wedge \bar{x} = Л$.

Основные законы алгебры Буля

1. Переместительный

$$x + y = y + x,$$
$$xy = yx.$$

2. Сочетательный

$$(x + y) + z = x + (y + z),$$
$$x(yz) = (xy)z.$$

3. Распределительный

а) раскрытие дизъюнкции по конъюнкции:

$$(x + y)\,z = xz + yz,$$

б) раскрытие конъюнкции по дизъюнкции:

$$(xy) + z = (x + z)(y + z).$$

4. Правило де Моргана:

$$\overline{(x + y)} = \bar{x} \bullet \bar{y},$$
$$\overline{xy} = \bar{x} + \bar{y}.$$

5. Исключение отрицания: $\bar{\bar{x}} = x$.

6. Исключение импликации:

$$x \to y = \bar{x} + y,$$
$$\overline{(x \to y)} = x\bar{y}.$$

7. Исключение эквивалентности

$$x \equiv y = (x \to y)(y \to x) = (\overline{x} + y)(x + \overline{y}).$$

8. Контрапозитивный закон

$$x \to y = \overline{y} \to \overline{x}.$$

9. Формулы склеивания

$$xy + x\overline{y} = x;$$

$$(x + y)(x + \overline{y}) = x.$$

10. Формулы поглощения

$$x + xy = x; \quad x(x + y) = x;$$

$$x + y\overline{x} = x + y; \quad x(\overline{x} + y) = xy.$$

В дальнейшем нам особенно часто придется пользовать распределительным законом, а именно - раскрытием конъюнкци по дизъюнкции. Чаще всего этот случай встречается в задач приведения исходной формулы к конъюнктивной нормальн форме. Рассмотрим маленький пример. (Для удобства чтен конъюнкцию иногда будем писать через точку).

Известно, что эквивалентность выражается формул $x \cdot y + \overline{x} \cdot \overline{y}$. Приведем ее к виду соотношения 7. Раскрываем п вую конъюнкцию xy по дизъюнкции: $(x + \overline{x} \cdot \overline{y})(y + \overline{x} \cdot \overline{y})$ и зат применяем этот закон внутри скоб $[(x + \overline{x})(x + \overline{y})][(y + \overline{x})(y + \overline{y})]$. Первая и последняя дизъюнкции тождества, т.е. равны $И$, в общей конъюнкции их можно отб сить (ибо $z \wedge И = z$). Окончательно:

$$xy + \overline{x} \cdot \overline{y} = (x + \overline{y})(\overline{x} + y)$$

5.5. Множество базовых аксиом

Как уже говорилось выше, в качестве базовых аксиом вы раются всегда общезначимые формулы. Их можно выбрать п разному. Десять подобных формул мы только что рассматрива в виде законов алгебры Буля.

В рамках исчисления высказываний в дополнении к этим конам предлагаются разные варианты систем аксиом.

Остановимся на следующей системе:

$$(\mathrm{A1}) \qquad A \to (B \to A);$$

$$(\mathrm{A2}) \qquad (A \to (B \to C)) \to ((A \to B) \to (A \to C));$$

$$(\mathrm{A3}) \qquad (\overline{A} \to \overline{B}) \to ((\overline{A} \to B) \to A).$$

Общезначимость (тождественность) всех этих постулатов легко жно проверить простой подстановкой различных значений A, B и . Но это займет много места. Поэтому покажем общезначимость лько первого постулата. Что же касается других, то в качестве имера они будут доказаны иными методами несколько позже.

Итак, первый постулат. Подставим в (A1) $B = И$, $A = Л$. То- , согласно таблице истинности, импликация $(B \to A) = Л$. Так к A ложно, получающаяся при этом вторая импликация типа $Л$ $И$ истинна.

Теперь введем интерпретацию $B = Л$ и $A = Л$, тогда импликация в скобках $(B \to A) = И$, а основная импликация принимает вид $И$, что дает опять-таки $И$.

Примем $A = И$ и $B = И$. Легко показать, что окончательная пликация здесь приходит к виду $И \to И$, т.е. равна $И$.

Наконец, положим $A = И$, $B = Л$. По таблице истинности (B) здесь равна $И$, а конечная ситуация $И \to И$ снова дает $И$.

Но тождественность постулатов не единственное условие, ко- ому система W (см. п. 4.2.2) должна удовлетворять. Необходи чтобы постулаты были *независимы*. Это значит, что ни один из нельзя вывести из двух оставшихся. Кроме того, система W жна быть еще и *полной*. Полнота системы означает, что любая ф, выводимая из этой системы, общезначима, и наоборот, любая значимая формула, полученная в рамках ИВ, - выводима из нной системы аксиом. Желательно при этом, чтобы система была *нимальна*, т.е. не содержала "лишних" аксиом, хотя очевидно, *присоединение к системе W любой общезначимой (и, следова льно, выводимой) формулы полноту системы не нарушает.*

5.6. Правила вывода

Напомним, что множество R (п. 4.2.2) содержит такие правила, торые позволяют на основании принятых постулатов определить тинность любых других выражений (ппф). Другими словами, они зволяют из тождественно истинных формул выводить другие то ественно истинные формулы. Эта процедура так и называется *вывод* (или иначе - *доказательство*). Полученные в результате щезначимые формулы называются *тавтологиями* или *теорема*

Если формула B выводима из формулы A, то пишут A Формулы же, принятые за аксиомы, обозначаются без антеце та: $\mapsto A$ (говорят: A выводится из пустого множества).

Основное множество R содержит два правила:

- правило подстановки и
- правило заключения.

Правило подстановки. Обратимся вновь к первому пост ту (A1). Он общезначим при всех интерпретациях. Заменим перь формулу A на формулу D. У этой формулы, так же, как A, только два возможных значения *И* или *Л*. Перепишем те постулат: $D \to (B \to D)$ - и что изменилось? При любых знач ях D постулат все равно общезначим.

Отметим теперь, что A - прописная буква, т.е. под ней п мается любая другая, в том числе и сложная, ппф. Пусть тепе заменяется на формулу $(a + b) \to r$. При различных интерпр циях a, b и r эта формула может принимать всего лишь два чения *И* или *Л*. Поэтому, подставив указанную ппф вместо A: $+ b) \to r) \to [B \to ((a + b) \to r)]$, мы, как и прежде, не наруш ее тождественности. Такие же рассуждения можно приве относительно B. Отсюда правило подстановки: **в любую *об* *значимую* формулу X, содержащую формулу Y, можно вм Y подставить любое другое высказывание (формулу) Z условии, что это сделано во всех местах вхождения Y. Ис ность X при этом не меняется.**

Оговорка насчет всех мест вхождения очень существе Посмотрите: D мы вводим вместо A в обоих случаях. Недопу мо заменять только одно A, а также одно A заменить на D, а гое на C. В общезначимой формуле $\overline{a} + a$ заменим одно a $a + r$. Полученная формула уже необщезначима! Но вполн пустимо заменять разные буквы на одни и те же: постулат (представленный в виде $C \to (C \to C)$, все равно общезначим.

Правило заключения. Это правило часто называют по-лат *modus ponens* (*MP*). Правило заключения нам уже известно. рассматривалось в рассказе об импликации как следстви

йств. Напомним суть: если импликация $p \to q$ истинна и p ис- ина, то и q тоже истинна. Иначе говоря, **если импликация ис- ина и антецедент истинен, то и консеквент тоже истинен**.

Пусть, например, имеется аксиома $\mapsto (A \to (B \to A))$, она все- и истинна. Если окажется, что при этом $A = И$, то по правилу *P* консеквент $(B \to A)$ будет также истинен.

Разберем теперь пример: пользуясь правилами вывода, пока- м, что формула $A \to A$ выводима из аксиом (A1) - (A3) и, сле- ательно, тавтология.

Возьмем постулат (A1): $A \to (B \to A)$ и сделаем в нем под- новку: вместо B подставим $(A \to A)$,получим:

1) $A \to ((A \to A) \to A)$.

Возьмем теперь (A2). Подставив $(A \to A)$ вместо B и A вме- C, получим

2) $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$.

тецедент формулы 2) совпадает с аксиомой 1) и потому исти- н. Применяя *modus ponens* к 1) и 2), получаем тавтологию:

3) $(A \to (A \to A)) \to (A \to A)$.

перь берем постулат (A1) и заменяем B на A:

4) $A \to (A \to A)$.

тецедент формулы 3) совпадает с 4) и, следовательно, истинен. именяя *MP*, получаем тавтологию:

5) $A \to A$.

5.7. Нормальные формы

Нормальные формы играют важную роль, поскольку позво- ют стандартизировать и упрощать процесс логического вывода ализа формул.

Формула исчисления высказываний имеет **нормальную рму, если она не содержит связок импликации (→) и экви лентности (≡) и знаки отрицания стоят в ней только при ременных.**

Освободиться от импликации и эквивалентности мы можем, именяя соотношения алгебры Буля 6 и 7, перенос и исключе ие отрицания - по правилу де Моргана и равенству 5 (стр. 56).

Все эти преобразования относятся к *равносильным*. Дл *формулы являются равносильными, если их таблицы истинно совпадают.*

Ниже предлагаются некоторые полезные равносильные отношения, следующие из прямого применения закона раскры конъюнкции по дизъюнкции. Их легко доказать самостоятел (Поскольку они следуют из законов Буля, мы продолжим их мерацию из п. 5.4.).

11. $ab + cd = (a + c)(a + d)\ (b + c)(b + d)$;
12. $abc + r = (a + r)(b + r)(c + r)$;
13. $(a + b)(x + y) + m = (a + b + m)(x + y + m)$;
14. $abc + p + q + r = abc + (p + q + r) = (a + p + q + r)(b + p$
$+r)(c + p + q + r)$.

Разберем теперь некоторые примеры. На них мы хотим по зать, как переход к нормальной форме упрощает доказательств

__Пример__ __1__. Привести выражение $ab \equiv \overline{(a \to \overline{b})}$ к нормаль форме и доказать его тождество.

$$ab \equiv \overline{(a \to \overline{b})};\quad ab \equiv \overline{(\overline{a} + \overline{b})};\quad ab \equiv \overline{\overline{ab}};\quad ab \equiv ab.$$

Далее применяем соотношение 7:

$$ab \equiv ab = (\overline{(ab)} + ab)(ab + \overline{(ab)}) = И.$$

__Пример__ __2__. Привести к нормальной форме и доказать истинно постулата (А3) системы аксиом ИВ

$$(\overline{A} \to \overline{B}) \to ((\overline{A} \to B) \to A).$$

Освобождаемся от импликации:

$$\overline{(A + \overline{B})} + (\overline{(\overline{A} \to B)} + A) = B\overline{A} + \overline{(A + B)} + A.$$

Далее:

$$B\,\overline{A} + \overline{A}\,\overline{B} + A = \overline{A}(B + \overline{B}) + A = И.$$

В дальнейшем, с целью упрощения, будем называть *лите* любое элементарное высказывание (или его отрицание).

Дизъюнктом будем называть дизъюнкцию некоторого ч литер:

$$(I_1 \vee I_2 \vee I_3 \vee ... \vee I_n) \text{ или } (I_1 + I_2 + I_3 + ... + I_n).$$

Любое логическое выражение (ппф) может быть представ либо в дизъюнктивной, либо в конъюнктивной нормальной фор *Дизъюнктивной нормальной формой* (ДНФ) называется дизъю ция конечного числа конъюнкций. *Конъюнктивной нормаль*

рмой (КНФ) называется конъюнкция конечного числа дизъюнк- в. Особый интерес к дизъюнктам объясняется их "наглядно- ью". В самом деле, достаточно лишь одному из "слагаемых" ъюнкта принять значение *И* ("1"), как весь дизъюнкт принимает начение *И* ("1"). Это очень удобно, когда решается вопрос о тож- ественности той или иной формулы. Переход от нормальной фор- ы выражения к его конъюнктивной нормальной форме в основ- ом осуществляется путем применения распределительного (дист- ибутивного) закона. Приведенные выше равносильные соотноше- ния 11-14 иллюстрируют примеры его применения.

Пусть задана формула $(p(s + r + q)) \to (p \to (s \to r))$.

Приводим ее к нормальной форме.

1. Исключаем импликации

$$\overline{(p(s + r + q))} + (\overline{p} + (\overline{s} + r)).$$

2. Переносим и снимаем отрицания (желательно, чтобы от- ицание относилось не более чем к одной литере)

$$(\overline{p} + \overline{(s + r + q)}) + (\overline{p} + \overline{s} + r);\quad (\overline{p} + (\overline{s} \cdot \overline{r} \cdot \overline{q})) + (\overline{p} + \overline{s} + r);$$

Некоторые скобки можно опустить:

$$(\overline{s} \cdot \overline{r} \cdot \overline{q}) + (\overline{p} + \overline{p} + \overline{s} + r).$$

Упрощая, получаем нормальную форму:

$$(\overline{s} \cdot \overline{r} \cdot \overline{q}) + (\overline{p} + \overline{s} + r).$$

3. Применяем распределительный закон: раскрываем конъ- юнкцию по дизъюнкции (ср. равносильность 14):

$$(\overline{s} + (\overline{p} + \overline{s} + r))(\overline{r} + (\overline{p} + \overline{s} + r))(\overline{q} + (\overline{p} + \overline{s} + r)).$$

Исключая внутренние скобки и пользуясь условием $\overline{r} + r = 1$, мечаем, что второй дизъюнкт равен 1 и, следовательно, исклю- ается из общей конъюнкции. Окончательно получаем КНФ $+ p + \overline{s})(\overline{q} + \overline{p} + \overline{s} + r)$.

Роль КНФ для оценки значений формул трудно переоценить. то возникает вопрос: нельзя ли установить какую-либо связь жду структурой формулы и ее семантикой? То есть связь меж- ее логической формой и логическим содержанием? Оказыва- ся, такая связь существует, и можно указать простой метод, по- ющий по виду формулы, приведенной к определенному ви- у, судить о том, тождественна она или нет. Метод как раз и заключается в приведении исходной формулы к нормальной кон юнктивной форме и последующем ее анализе.

Пусть имеется некоторое выражение F, представленное КНФ, $F = (a + b + r)(a + \bar{r} + b + s + r)(b + r + s + \bar{s})$. Сразу видно, ч второй и третий дизъюнкты равны 1, поскольку имеют по д одинаковые переменные, одна из которых - отрицание друго Эти дизъюнкты можно опустить. Но сама формула $F = (a + b +$ - необщезначима и не противоречива, т.е. выполнима.

Пусть теперь формула F имеет другой вид: $F = (a + b + p)r(b + \bar{p})(b + p + r)\bar{r}$. В ее составе два однолитерн дизъюнкта (r и $\bar{r}$) причем один их них является отрицани другого. Такие литеры называются *контрарными*. Конъюнкц контрарных литер равна 0, и поэтому мы имеем невыполниму функцию:

$$F = 0\ (Л).$$

Общий вывод. *Для того, чтобы некоторая КНФ-форму была общезначима, необходимо и достаточно, чтобы каждый ее дизъюнктов содержал хотя бы одну пару контрарных лит Для того, чтобы некая КНФ была невыполнима, необходим чтобы она содержала хотя бы одну пару однолитерных ко трарных дизъюнктов.*

5.8. Свойства ИВ как аксиоматической системы

То, что ИВ является аксиоматической системой, следует всего вышесказанного, так сказать, по построению. Здесь мы х тим лишь подчеркнуть некоторые свойства, присущие ИВ к всякой системе подобного рода.

В основе аксиоматической системы лежит формальная те рия. Что это означает, мы видели выше. Это означает, в части сти, что в системе определено некоторые множество W базов аксиом, из которых через процедуру вывода по правилам R мо но вывести некоторую формулу B, обязательно общезначиму ное множество W общезначимо по определению, а правила вы да не меняют истинности. Поскольку B общезначимо, то, очев но, ее отрицание $\bar{B} = Л$, т.е. невыполнимо и невыводимо из W.

Этот факт говорит о непротиворечивости системы: **система *непротиворечива*, если в ней невыводимы никакие две формулы, одна из которых является отрицанием другой.**

Ранее мы определили понятие полноты системы аксиом, согласно которому любая общезначимая ппф может быть выведена из этой системы и, следовательно, ее присоединение к этой системе полноту последней не нарушает. Покажем теперь, что полнота и непротиворечивость связаны между собой: **нарушение полноты приводит к нарушению непротиворечивости системы.**

Дизъюнкция $A + B$, например, не является общезначимой ппф и потому невыводима из принятой ранее системы аксиом (А1) - (А3). Предположим, однако, что она выводима и, следовательно, может быть добавленной к системе в виде четвертого постулата. Далее рассуждаем следующим образом.

Представим дизъюнкцию $A + B$ в виде импликации:

1) $\bar{A} \to B$,

так как она теперь считается общезначимой, проводим ряд подстановок и получаем "тавтологии":

2) $\bar{A} \to \bar{B}$ (заменив B на $\bar{B}$),

3) $A \to B$ (заменив в 1) A на $\bar{A}$).

Правило *modus ponens* может быть теперь применено к постулату (А3), т. к. его посылка ($\bar{A} \to \bar{B}$) теперь общезначима согласно тождеству 2). Отсюда справедливо тождество:

4) ($\bar{A} \to B) \to A$.

Применив *MP* к тождествам 1) и 4), имеем:

5) $A = И$.

Заменим в 4) A на $\bar{A}$:

6) $(A \to B) \to \bar{A}$.

Применяя теперь *MP* к 6) с учетом 3), получаем неопровержимо:

7) $\bar{A} = И$.

Сравнивая 7) и 5), отмечаем явное противоречие. Отсюда определение: **система *полна*, если присоединение к аксиомам какой-либо невыводимой формулы делает ее противоречивой**

5.9. Проблема логического вывода

Ранее мы говорили о выводе общезначимых формул (тавтоло

гий, тождеств) из базовой системы аксиом. Рассмотрим вопрос
вода шире. Пусть имеется множество необщезначимых формул
E_2, E_3...En, обладающих тем свойством, что при некоторых ин
претациях они одновременно принимают значение *И*. Формул
выводима из множества *{E}*, если она может быть выведена из
путем применения правила заключения. (Правило подстано
применяется только для общезначимых формул). Технически
цесс вывода будет заключаться в получении ряда последоват
ных формул B_1, B_2, B_3..., каждая из которых выводится из все
плюс те Bn, что уже получены прежде. Процесс длится до тех
пока одна из очередных Bk не совпадет с B, т.е. когда $Bk = B$.

Правило заключения не меняет истинности, и потому спра
ливо утверждение, что если B выводима из *{E}*, то она должна п
нимать значения *И* при тех же интерпретациях, что и все Ei. (
обще-то формулы Ei могут принимать самые разные значения. Р
идет о значениях истинности, общих для всех них и B в том числ

Пишут: $(E_1, E_2, \dots En)\mapsto B$ или короче $\{E\}\mapsto B$.

Говорят также, что B есть *логическое следствие* из посы
{E}. Вывод из системы аксиом теперь можно рассматривать
частный случай, когда выполняется условие $\mapsto\{A\}\mapsto B$.

Пример. Знакомое нам правило заключения можно пере
сать следующим образом: $(p, p\to q)\mapsto q$, т.е. q - логическое сл
ствие посылок p и $p\to q$. Справедливость этого легко провер
ся по таблице истинности: высказывание q принимает знач
И, как только p и $(p\to q)$ одновременно принимают значение

Проблема вывода сводится, таким образом, к *проблеме
дукции*, которая формулируется следующим образом:

*требуется определить, является ли формула B логичес
следствием множества формул {E}.*

Чтобы подчеркнуть, что Ei не являются аксиомами, их
гда обозначают через Hi и называют гипотезами, а формулу
заключением:

$$(H_1, H_2, \dots Hn)\mapsto B.$$

Если гипотезы рассматривать как посылки в рассуждении
из определения логического следствия следует, что заключе
истинно только тогда, когда все посылки истинны. Заметим

что если для множества *{E}* не существует такой интерпрета-
при которой все Ei принимают значения *И*, то множество
считается невыполнимым и обозначается:

$$(E_1, E_2, \dots En)\mapsto Л \text{ или } (H_1, H_2, \dots Hn)\mapsto Л. \qquad (5.2)$$

разу же отметим одно важное свойство, следующее из оп-
ления логического следствия. Если $\{E\}\mapsto B$, то B принимает
чение *И* как только все E_i также принимают значение истин-
Исключается, таким образом, случай, когда $B=Л$ при всех
(обозначим: $\{E\}=И$). Остальные варианты вполне допусти-
ведем их в табличку, добавив столбец с импликацией, кото-
нам сейчас пригодится.

Таблица 3

{E}	*B*	*{E}→B*
И	*И*	*И*
Л	*И*	*И*
Л	*Л*	*И*
И	*Л*	*Исключ.*

таблицы 3 следует очевидный и важный вывод: если B -
ческое следствие из множества *{E}*,то импликация $\{E\}\to B$ -
ственно истинна.

словие одновременной истинности всех Ei соответствует
ванию конъюнкции. Следовательно, можно написать:

$$(E_1 \wedge E_2 \wedge E_3 \dots \wedge En)\to B \mapsto И$$

иначе $$(H_1 \wedge H_2 \wedge H_3 \dots \wedge Hn)\to B \mapsto И. \qquad (5.3)$$

Вопрос, таким образом, сводится к определению тождест
ности выражения (5.3).

Пример. Известное нам правило заключения может быть за
но следующим образом:

a) $(p, p\to q)\mapsto q$.

Если q - логическое следствие из p и $p\to q$, то выражение

b) $(p \wedge (p\to q))\to q$

но быть тождеством.

Покажем это. Приведем ппф b) к конъюнктивной нормальной
форме (дизъюнкцию будем обозначать +).

Освободимся от импликаций и преобразуем:

$$\overline{\left(p\left(\overline{p}+q\right)\right)}+q$$

$$\overline{p}+\overline{\left(\overline{p}+q\right)}+q;$$

$$\overline{p}+\left(p\overline{q}+q\right);$$

c) $\left(\overline{p}+q\right)+\left(p\overline{q}\right)$.

К выражению c) применим распределительный закон р крытия конъюнкции по дизъюнкции:

d) $\left(p+\overline{p}+q\right)\left(p+q+\overline{q}\right)\mapsto И$.

Полученная конъюнктивная форма d), очевидно, тожде венна, поскольку каждый из двух ее дизъюнктов равен *И*, ($p+\overline{p}$=*И*, $q+\overline{q}$=*И*). Что и требовалось доказать.

В данном примере мы вновь встретились с задачей опреде ния тождественности формулы. *В* более общей формулировке будет встречаться нам и в дальнейшем: требуется определит какому из трех классов принадлежит данная формула, являе ли она: а) тождественной, б) невыполнимой или в) выполним В этом состоит *проблема разрешения*.

5.10. Алгоритмическая проблема разрешения в ИВ

Задача, состоящая в отыскании процедуры, позволяющей любой формулы выяснить, к какому из трех вышеназванных клас она принадлежит, называется еще *семантической проблемой ра шения*. В соответствии с этим процедура, позволяющая конеч числом простых действий решить проблему разрешения, называ разрешающей процедурой. Самое естественное решение здесь - ратиться к таблице истинности. Таблица даст исчерпывающий о о поведении формулы при всех возможных интерпретациях. Но прием малоэффективен, поскольку практически он применим л для малого числа аргументов (литер). Если формула из двух л имеет 4 варианта интерпретаций (т.е. 4 строки), то формула из че рех литер уже 16, из пяти соответственно 32 и т.д. по степени 2^{n} ст таблицы истинности. Кстати, этот прием мы применили, когда д вывали тождественность первого постулата ИВ (A1).

Заметим, однако, что для того, чтобы получить разреш щую процедуру, достаточно найти способ, позволяющий о чить тождественные формулы от всех остальных. Применяем

оцедуру к некоторой формуле *A* и, если окажется, что *A* обще- начима, то проблема решена. Если же выясняется, что *A* не об- значима, то применяем эту процедуру для формулы $\overline{A}$. Если окажется тождественной, то, очевидно, *A* - противоречива. Ес- $\overline{A}$, так же, как и *A*, не тождественная, то это уже значит, что ормула *A* просто выполнима.

Существует несколько методов оценки тождественности ормулы:

1) оценка с помощью таблицы истинности,

2) оценка через преобразование, упрощение и приведение к ормальным формам,

3) оценка путем логического вывода из системы аксиом,

4) оценка методом редукции,

5) оценка методом опровержения.

Некоторые из этих методов мы уже "прошли", с другими пред- оит познакомиться. Так, например, в п. 5.5 мы проводили оценку ождественности постулата (A1) с помощью таблицы истинности. нализу значений нормальных форм был посвящен п. 5.7. В п. 5.6 рогим логическим выводом доказали тавтологию $A \to A$.

В ряде случаев для определения тождеств удобен так назы- емый *алгоритм редукции*. Алгоритм основан на доказательстве утем приведения к абсурду. Метод особенно хорош, когда фор- улы содержит много импликаций. Рассмотрим его на примере орого постулата (A2) исчисления высказываний.

Предположим невероятное, что (A2) ложен, и докажем урдность этого предположения.

Итак, пусть:

1) $[(A \to (B \to C)) \to ((A \to B) \to (A \to C))] = Л$.

По свойству импликации, она ложна, если антецедент исти ен, а консеквент - ложен, т.е.

2) $(A \to (B \to C) = И$,

3) $((A \to B) \to (A \to C)) = Л$.

Тогда из равенства 3) следует:

4) $(A \to B) = И$,

5) $(A \to C) = Л$.

Ложность $(A \to C)$ означает, что $A = И$, $C = Л$.

Применив *modus ponens* к $(A \to B)$, получаем: если $A = И$, то
$= И$. Подставим теперь полученные значения A, B и C в 2), получи
6) $(A \to (B \to C)) = (И \to (И \to Л) = Л$,
что противоречит ранее полученному значению 2). Мы пришли
абсурду и тем самым показали, что исходный постулат общезначи

К сожалению, алгоритм редукции не всегда удобен в при
нении и не может быть рекомендован как универсальный ме
определения тождественности.

5.11. Теорема дедукции

Проблема разрешимости тесно связана с проблемой выво
Методов много, но тем не менее процесс этот остается достат
но утомительным даже для простых примеров. Проблема выв
существенно упрощается, если применить *теорему дедукц*
Пусть имеется множество гипотез $\{H\}$ и Hn - одна из них.

Теорема утверждает, что

необходимым и достаточным условием выводимости B
гипотез $\{H\}$ *является выводимость импликации* $(H_n \to B)$,
есть из утверждения $(H_1, H_2, \ldots H_n) \mapsto B$ *следует:*

$$(H_1, H_2, \ldots H_{n-1}) \mapsto (H_n \to B). \qquad (5.4)$$

Пусть это условие выполняется и $(H_n \to B)$ выводимо. О
значим эту формулу через $M1$. Тогда по условию теоремы долж
быть выводимой формула $(H_{n-1} \to M1)$. Обозначим ее через M
теперь будет выводима формула $(H_{n-2} \to M2)$ и т.д. Формулиро
теоремы, следовательно, сводится к следующей:

Если B *выводима из гипотез* $\{H\}$, *то истинно выражение*

$$(H_1 \to (H_2 \to (H_3 \to \ldots (Hn \to B))\ldots). \qquad (5.5)$$

Необходимость утверждения доказывается следующим об
зом. Допустим, B истинна, тогда импликация $(H_n \to B)$ также ист
на при любой H_n. Но это, в свою очередь, означает, что истинн
следующая импликация $(H_{n-1} \to (H_n \to B))$ и т.д. вплоть до заклю
тельной импликации $(H_1 \to (H_2 \to \ldots.)\ldots)$.Что и требовалось доказа

Выражение типа (5.5) с "вложенными" импликациями на
вается n-кратной импликацией. Доказав необходимость (5.5), м
вывели такое логическое утверждение:

если в n-кратной импликации заключение B истинно, то

й *вероятности* ее посылки Hi тоже истинны.

Докажем, что это так и есть, т.е. докажем теперь достаточ
сть. Она доказывается простым применением правила заклю
ния. В самом деле, положим, что выражение (5.5) истинно.

Если H_1 принимает значение $И$, то по *modus ponens* истинно и
выражение $(H_2 \to (H_3 \to \ldots)\ldots)$. Но при той же интерпретации
инна и H_2, откуда следует, что истинно и выражение $(H_3 \to$
$H_4 \to \ldots)\ldots)$. И т.д. В конце цепочки рассуждений приходим к вы
ду, что истинна импликация $H_n \to B$ (выполнение условия (5.4)
оремы). Но H_n тоже гипотеза и тоже истинна.

Отсюда $B = И$.

Рассмотрим примеры.

а. Доказать свойство транзитивности импликации:
из условия $(A \to B)$ *и* $(B \to C)$, *следует* $(A \to C)$.

Вопрос сводится к доказательству логического следствия:

$$(A \to B,\ B \to C) \mapsto (A \to C). \qquad (5.6)$$

Доказательство здесь неочевидно, но согласно теореме де
укции нам достаточно доказать условие:

$(A \to B,\ B \to C,\ A) \mapsto C$.

Итак, имеем гипотезы:

a. H_1: $A \to B$,
b. H_2: $B \to C$,
c. H_3: A.

Применяя MP к a и c, получаем

d. B и далее:
e. C (*modus ponens* к b и d).

Что и требовалось.

Доказать выражение (5.6), пользуясь условием (5.5).

Гипотезы у нас уже обозначены. Подставляем их в (5.5):

$$((A \to B) \to ((B \to C)\to (A \to C))). \qquad (5.7)$$

Импликация $A \to C$ является логическим следствием, если
принимает значение $И$ тогда, когда все $Hi = И$. Пусть форму
(5.7) истинна. Если теперь положим, что $(A \to B) = H_1$ истинна,

то по MP $((B \to C) \to (A \to C)) = И$.

Но $(B \to C) = H_2$ тоже истинна, и тогда $(A \to C) = И$ и потому является логическим следствием по условию (5.6).

5.12. Принцип дедукции

Вернемся к выражению (5.1) $(H_1, H_2, ... Hn)\mapsto B$ и предпо жим, что оно выполняется, т.е. B - выводимо из *{H}*. Если это то присоединение B к множеству *{H}* не влияет на полноту м жества *{H, B}* и не приводит его к противоречию. Иное дело которая при этих условиях невыводима. Ее присоединение к очевидно, приведет к противоречивости, т.е. можно написать:

$$(H_1, H_2, ... Hn, \overline{B}) \mapsto Л.$$

При этих условиях тождественность импликации (5.3), с ственно, нарушается, она становится невыполнимой. Возьмем рицание импликации (5.3). Теперь должно выполняться услов

$$\overline{((H_1 \wedge H_2 \wedge H_3 ... \wedge H_n) \to B)} \mapsto Л\ .$$

Доказав невыполнимость (5.8) или (5.9), мы косвенно дока выполнимость условия (5.1). Такой метод доказательства "от тивного" называется *доказательством по методу опровержени*

Метод опровержения очень удобен. В самом деле, вмест го, чтобы кропотливо доказывать общезначимость какой-л формулы F, достаточно доказать невыполнимость ее отриц $\overline{F}$,что в ряде случаев значительно проще. (В 60-х годах XX был открыт и разработан так называемый принцип резолю весьма подходящий для доказательств подобного рода).

Метод опровержения реализует *принцип дедукции*, кот формулируется следующим образом:

формула B является логическим следствием множеств тогда и только тогда, когда множество $\{H, \overline{B}\}$ *невыполн т.е.* $(H_1, H_2, ... H_n) \mapsto B$, *если* $(H_1, H_2, ... H_n, \overline{B}) \mapsto Л$.

Докажем условие (5.9).

Исключим импликацию:

$$\overline{\overline{(H_1 \wedge H_2 \wedge ... \wedge H_{n1})} \vee B}$$

и по правилу де Моргана получим:

$$\overline{\overline{(H_1 \wedge H_2 \wedge ... \wedge H_n)}} \wedge \overline{B}\ ,$$

что эквивалентно выражению

$$H_1 \wedge H_2 \wedge ... \wedge Hn \wedge \overline{B}.$$

Дело, таким образом, сводится к доказательству невыполнимости

$$(H_1 \wedge H_2 \wedge ... \wedge Hn \wedge \overline{B}) \mapsto Л. \qquad (5.11)$$

Формула (5.10) в своем виде уже является элементом КНФ, же если все Hi - элементарные высказывания. Если же Hi - ожные выражения, то они приводимы к виду КНФ и их подста- нки в (5.10) не меняет ее нормальной формы. Поэтому в даль- шем мы вправе считать каждое из Hi дизъюнктом, состоящим одного или многих "слагаемых".

Условие (5.11) требует, чтобы формула (5.10) была невыпол- мы, т.е. тождественно равна 0. Это возможно, если среди ее множителей" найдется хотя бы один пустой дизъюнкт. Такой зъюнкт не может появиться в явном виде. Но он может быть ыделен на множестве $\{H, \overline{B}\}$ путем определенных эквивалент- ых преобразований.

Из всего вышесказанного следует, что согласно принципу дукции вопрос о выводимости (невыводимости) некоторой рмулы B сводится, в конечном счете, к анализу невыполнимо- и множества дизъюнктов. Но множество дизъюнктов невыпол- мо тогда и только тогда, когда пустой дизъюнкт $Л$ является ло- ческим следствием из него. Таким образом, невыполнимость ножества $\{H, \overline{B}\}$ можно проверить, порождая логические след- вия из него до тех пор, пока не получится пустой дизъюнкт. тод, позволяющий получить логические следствия из множе- ва дизъюнктов, основан на применении принципа резолюций.

5.13. Принцип резолюций

В сущности в основе принципа лежит несложная схема рассуждений. Пусть A, B и C - формулы и имеются два юнкта: C) и 2. $(B + \overline{C}.:)$,

торые мы будем считать истинными. Положим теперь, что $C = И$.

Подставив это значение в первое выражение, получим дизъ- нкт вида $A + И$, который истинен при любом A. Подставив $C = И$ второе выражение, получим дизъюнкт $B + Л$, из которого од- значно следует, что $B = И$, поскольку принято, что весь дизъ- нкт истинен.

Положим теперь $C = Л$. Второй дизъюнкт истинен при любом

B, но первый принимает вид $A + Л$, откуда следует, что $A = И$.

Все это означает, что, независимо от интерпретации формулы C, либо A, либо B истинны. Это можно отразить в виде нового дизъюнкта $(A + B)$, исключив *контрарные формулы* C *и* $\overline{C}$.

Другими словами, выполняется правило:

$$(A + C, B + \overline{C}) \mapsto (A + B). \qquad (5.12)$$

Правило особенно эффективно, если A и B дизъюнкты, а C высказывание. Это правило называется *правилом резолюции*, а вновь полученный дизъюнкт - *резольвентой*. Он формируется как "сумма" оставшихся формул, за исключением *контрарных*, т.е. разных по знаку.

Для дизъюнктов $(p + q + \overline{r})$ и $(m + r)$, например, резольвента будет иметь вид $(p + q + m)$. Для дизъюнктов $(p + r)$ и $(\overline{r})$ соответственно (p), а вот в случае (p) и $(\overline{p})$ обе литеры "уничтожаются". Это пример получения *пустого* дизъюнкта. Еще один пример. Имеются два дизъюнкта $(\overline{x} + y)$ и $(x + \overline{y})$. Получаемые здесь резольвенты $(x + \overline{x})$ или $(y + \overline{y})$ равны 1, информации не несут, их следует отбрасывать.

В логическом плане каждый полученный дизъюнкт - резольвента равносилен обоим дизъюнктам-родителям, участвовавшим в резолюции, (они так и называются - родители). Он становится очередной гипотезой и участвует в резолюции на равных с другими. Это касается и пустого дизъюнкта, но будучи поставлен в ряд гипотез Hi, он делает формулу типа (5.10) равной 0, что и является ее доказательством.

<u>*Пример.*</u> Доказать невыполнимость множества дизъюнктов: $H = (p + q, p + r, \overline{q} + \overline{r}, \overline{p})$. Пронумеруем гипотезы:

1. $p + q$,
2. $p + r$,
3. $\overline{q} + \overline{r}$,
4. $\overline{p}$.

Далее применяется правило резолюции. Полученные резольвенты присоединяются к исходному множеству, с ними снова можно проводить резолюции. Ниже следует список резольвент, в скобках указываются номера резольвент, участвовавших в резолюции.

5. q (1,4),
6. r (2,4),
7. $\overline{q}$ (3,6),
8. $Л$ (5,7).

Надо отметить, что выбранный путь порождения резольвент может быть не единственным. Например, в нашем случае результат можно было получить чуть быстрее, если пойти по следующему пути:

5. $p + \overline{r}$ (1,3),
6. p (2,5),
7. $Л$ (4,6).

Как видно из примера, метод резолюций легко представим в виде несложного регулярного алгоритма. Это предопределяет успешное применение ЭВМ в решении задач вывода, хотя здесь и возможны неожиданные трудности. Дело в том, что машина ищет резолюцию слепо, методом перебора. При таком поиске путь к результату может быть весьма долгим. Возможны случаи простого зацикливания машины. Простейший пример: имеется два дизъюнкта p и $\overline{p}+q$. При машинной реализации резольвента q может порождаться неограниченное число раз. Предусматривая подобные случаи, применяют специальные стратегии поиска резольвент и соответствующее программирование. Некоторые такие стратегии мы рассмотрим в следующем разделе.

В заключение отметим такое важное обстоятельство: множество резольвент (вместе с родительскими дизъюнктами) образует не что иное, как математическую модель некоторой предметной области. Каждая новая резольвента добавляет новое состояние в пространство состояний *Мпо*, а поиск решения в пространстве состояний представлен операцией логического вывода.

5.14. Свойства метода резолюций

Несмотря на известные недостатки, заключающиеся главным образом в возможности неограниченного порождения резольвент и в отсутствии тем самым гарантий быстрого поиска решения, метод резолюций является одним из самых мощных инструментов в решении задач логического вывода. Он обладает двумя существенными достоинствами: он логичен, т.к. резольвенты являются логическими следствиями предложений-родителей, и он обладает полнотой, т.к. при использовании в процессе вывода результат получается за счет применения только одного и того же правила. Представляет интерес свойство *завершаемости* метода резолюций: если множество *{H}* невыполнимо, то пустой дизъюнкт может быть найден посредством резолюций. Это и понятно: пустой дизъюнкт есть резольвента множества *{H}* и, будучи невыполнимым, не может быть следствием выполнимого множества. По этому поводу даже есть лемма: *если множество {H} невыполнимо и содержит резольвенты своих элементов, то оно обязательно содержит пустой дизъюнкт.*

Метод резолюций применяется не только в рамках исчисления высказываний и исчисления предикатов (см. последующие главы), к нему нередко сводится процедура вывода в некоторых других моделях представления знаний (продукционные ПЗ, семантические сети, фреймы). Широко известный язык программирования ПРОЛОГ специально создан для реализации процедур, основанных на методе резолюций. Что же касается указанных недостатков, то в значительной степени они уменьшаются или даже исключаются путем применения специальной стратегии целенаправленного поиска резольвент. Ниже мы коснемся этого подробнее.

Основное назначение метода резолюций - порождение логических следствий из системы гипотез *{H}* (причем эта система должна быть представлена в конъюнктивной нормальной форме). Это может быть порождение с целью общего обзора следствий, или же это будет поиск пустого дизъюнкта при реализации принципа дедукции - меняется только цель. Резольвента есть логическое следствие своих родителей. Обратного утверждать нельзя!

Рассмотрим показательный пример. Обратимся все к тому же свойству транзитивности импликации: если $(a \to b)$ и $(b \to c)$, то $(a \to c)$. Иначе: $[(a\to b)(b \to c)] \mapsto (a \to c)$.

Гипотезы в нашем примере: H_1: $(a \to b)$ и H_2: $(b \to c)$, формулы B: $(a \to c)$. Согласно принципу дедукции, условие выполняется, если показано, что $(H_1, H_2, \overline{B}) \mapsto Л$. Переходя к КНФ, получаем выражение $[(\overline{a}+b)(\overline{b}+c)(\overline{\overline{a}+c})] \mapsto Л$. Задача, таким образом, сводится к знакомой формулировке: доказать невыполнимость множества дизъюнктов: $(\overline{a} + b)$, $(\overline{b} + c)$, a, $\overline{c}$. (т.к $\overline{(\overline{a}+c)} = a \wedge c$ - это два однолитерных дизъюнкта a и $\overline{c}$).

Дизъюнкты:	Резольвенты:
1. $\overline{a}+b$,	5. b (1,3),
2. $\overline{b}+c$,	6. c (2,5),
3. a,	7. $Л$ (4,6).
4. $\overline{c}$.	

Транзитивность доказана, и "по логике" это легко понять: если из a следует b, а из b следует c, то очевидно, что из a следует c. Интересно, будет ли логическим следствием из указанных дизъюнктов какой-либо другой из них? Например, $\overline{b} + c$. То есть требуется доказать:

$$[(a \to b)(a \to c)] \mapsto (b \to c).$$

Переходя к КНФ, имеем дизъюнкты: $(\overline{a} + b)$, $(\overline{a} + c)$, $\overline{(\overline{b}+c)}$ или иначе $(\overline{a} + b)$, $(\overline{a} + c)$, b, $\overline{c}$.

Итак, дизъюнкты:	Резольвенты:
1. $\overline{a} + b$,	5. $\overline{a}$. (2,4)
2. $\overline{a}+ c$,	Других резольвент нет.
3. b,	Импликация $(b\to c)$ не является
4. $\overline{c}$.	логическим следствием.

Оно и правда: из того, что b и c следуют из a вовсе не следует, что c следует из b. Из примера выводим:

если резольвента Bi есть логическое следствие из {H}, то совсем не обязательно, чтобы хотя бы для одной гипотезы H_k выполнялосьловие: $(H_1, H_2, .. H_n, B_i) \mapsto Hk$.

Разберем еще несколько примеров.

__Пример__ 1. Пусть даны посылки p и $p \to q$. Определить все возможные следствия из этой пары. Систематический обзор логических следствий удобно осуществлять путем обращения к *совершенной* конъюнктивной нормальной форме (СКНФ). Напомним, что особенность СКНФ заключается в том, что все ее конюнктивные члены содержат все переменные данной формулы и, следовательно, имеют одинаковое число литер. Если какой-либо член СКНФ содержит меньшее число переменных, то недостающие переменные вводятся прибавлением формулы типа $a \wedge \bar{a}$, которая равна 0 и на дизъюнкцию не влияет. Затем эту конъюнкцию раскрывают по дизъюнкции. Все это мы сейчас проделаем на примере. Как и прежде, образуем конъюнкцию $p(p \rightarrow q)$ и приводим ее к СКНФ. Освобождаемся от импликации: $p(\bar{p} + q)$. В дизъюнкте p не хватает члена q. Прибавляем конъюнкцию $\bar{q} \wedge q$: $(p + (\bar{q} \wedge q))(\bar{p} + q)$ и, раскрывая её, имеем: $(p + q) \wedge (p + \bar{q}) \wedge (\bar{p} + q)$. Полученная СКНФ дает нам не только три конъюнктивных члена: $(p + q)$, $(p + \bar{q})$ и $(\bar{p} + q)$, каждый из которых является логическим следствием, но и возможность получать другие следствия в виде конъюнкции любой пары из них:

$(\bar{p} + q) \wedge (p + \bar{q})$, $(p + q) \wedge (\bar{p} + q)$, $(p + q) \wedge (p + \bar{q})$.

Но и это еще не все следствия. Применяя метод резолюции, получаем дополнительно дизъюнкты: p и q (Проверить самостоятельно).

<u>*Пример*</u> 2. Институт заключает договор, в котором предположительно могут участвовать три кафедры *A, M* и *R*. Однако участие этих кафедр обусловливается рядом обстоятельств. А именно:

а) если *M* не участвует в договоре, то не участвует и *A*;

б) если же *M* участвует, то участвуют и *A*, и *R*.

Но вот вопрос: обязана ли *R* участвовать в договоре, если в нем уже участвует *A*?

Опишем задачу в терминах исчисления высказываний. Обозначим: a – в договоре участвует *A*, m - в договоре участвует *M*, и r – в договоре участвует *R*. Тогда условию а) соответствует формула $\bar{m} \rightarrow \bar{a}$, условию б): $m \rightarrow (a \wedge r)$. Необходимо решить, следует ли из этих условий $a \rightarrow r$?

Требуется вывести логическое следствие:

$((\bar{m} \rightarrow \bar{a}), (m \rightarrow (a \wedge r)) \mapsto (a \rightarrow r)$.

Исключаем импликации и делаем небольшие преобразования:

$[(m + \bar{a}), (\bar{m} + ar)] \mapsto (\bar{a} + r)$,

$[(m + \bar{a}), ((\bar{m} + a)(\bar{m} + r))] \mapsto (\bar{a} + r)$,

$[(m + \bar{a}), (\bar{m} + a), (\bar{m} + r)] \mapsto (\bar{a} + r)$.

Согласно принципу дедукции, требуется доказать условие:

$[(m + \bar{a}), (\bar{m} + a), (\bar{m} + r), \overline{(\bar{a} + r)}] \mapsto Л$.

Применяя метод резолюции, анализируем:

дизъюнкты:

1. $m + \bar{a}$,
2. $\bar{m} + a$,
3. $\bar{m} + r$,
4. a,
5. $\bar{r}$.

резолюции:

6. m, (1,4)
7. r, (3,6)
8. $Л$. (5,7)

Кафедра *R* обязана участвовать
в договоре, если участвует *A*.

5.15. Пример решения задачи средствами ИВ

Обратимся к известной уже нам задаче про обезьяну и бананы (п. 1.5). На основе фактов БД и правил БЗ построим формальную модель предметной области на языке *исчисления высказываний* и проиллюстрируем на ней механизм логического вывода.

Сначала для этого необходимо определить алфавит, т.е. набор символов, обозначающих высказывания, которые мы определим, например, следующим образом:

A – "Обезьяна находится в точке *a*",

B – "Ящик находится в т. *b*",

C – "Бананы находятся в т. *c*",

D – "Обезьяна находится на Ящике",

E – "Обезьяна держит Бананы",

F – "Обезьяна, Ящик, Бананы находятся в разных точках",

G – "Обезьяна находится рядом с ящиком".

Используя логические связки: ¬, ∧, ∨, →, ≡, мы можем строить более сложные умозаключения. Например:

1) $(A \wedge B \wedge C) \rightarrow F$.

(Если все предметы находятся на своих исходных позициях, е, в точках *a*, *b*, *c* соответственно, то справедливо сказать: Обезьяна, Ящик, Бананы находятся в разных точках").

2) $F \rightarrow \bar{G}$.

(Если "Обезьяна, Ящик, Бананы находятся в разных точках", о "Обезьяна находится не рядом с Ящиком").

3) $F\to \bar{D}$.

(Если "Обезьяна, Ящик, Бананы находятся в разных точках", то "Обезьяна не находится на Ящике").

4) $F\to \bar{E}$.

(Если "Обезьяна, Ящик, Бананы находятся в разных точках", то "Обезьяна не держит Бананы").

Высказывания 2, 3, 4 можно объединить в одно с помощью логической связки "*И*":

5) $F\to(\bar{G} \wedge \bar{D} \wedge \bar{E})$.

6). $\bar{F}\to(D\wedge E)$.

(Если "Обезьяна, Ящик и Бананы находятся не в разных точках, (т.е. в одной), то "Обезьяна стоит на Ящике" ***И*** "Обезьяна держит Бананы").

7) $\bar{D}\to \bar{E}$.

(Если "Обезьяна не на Ящике", то "Обезьяна не держит Бананы").

Таким образом, фрагмент предметной области на языке исчисления высказываний (ИВ) представляет собой следующий набор формул (БЗ):

$$
\begin{aligned}
&1.\ (A\wedge B\wedge C)\to F;\\
&2.\ F\to(\bar{G}\wedge \bar{D}\wedge \bar{E});\\
&3.\ \bar{F}\to(D\wedge E);\\
&4.\ \bar{D}\to \bar{E}.
\end{aligned}
\qquad (5.13)
$$

Для простоты введем обозначение $S = A\wedge B\wedge C$.

Теперь, используя *механизм логического вывода* ИВ, мы можем доказать выводимость любой другой формулы, структура которой соответствует синтаксису ИВ (в рамках данной модели). Докажем, например, что формула:

$$(\bar{S}\wedge D)\to E \qquad (5.14)$$

выводима. Ее смысл: если "Обезьяна, Ящик и Бананы не в исходных точках" ***И*** "Обезьяна на Ящике", то "Бананы в руках у Обезьяны".

Логический вывод можно сделать несколькими способами.

1) Можно, например, использовать *таблицу истинности*, которая дает исчерпывающую картину значений переменных. При этом формулы (5.13) и (5.14) образуют выражение логического вывода в виде:

$$((A\wedge B\wedge C)\to F)\wedge(F\to(\bar{G}\wedge \bar{D}\wedge \bar{E}))\wedge(\bar{F}\to(D\wedge E))\wedge(\bar{D}\to \bar{E}))\mapsto$$
$$((\bar{S}\wedge D)\to E). \qquad (5.15)$$

Далее следует вычислить значения выражений слева и справа от знака $\mapsto$. Если окажется, что формула справа принимает значение *И* как только все формулы (5.13), образующие левую часть, одновременно примут значение *И*, то логическая выводимость (5.14) из (5.13) доказана.

Учтем, однако, следующее. Формула (5.15) имеет 7 переменных. Это значит, что таблица будет содержать 2^7, т.е. 128 строк! Есть ведь и другие методы.

2) Можно попытаться использовать метод вывода, основанный на *системе аксиом исчисления высказываний*. Но это тоже очень трудоемкий процесс.

3) Лучше всего попробовать метод опровержения, основанный на *принципе дедукции* и реализуемый посредством *резолюций*. Согласно этому методу следует доказать противоречивость системы

$$
\begin{aligned}
&1)\ S\to F;\\
&2)\ F\to(\bar{G}\wedge \bar{D}\wedge \bar{E});\\
&3)\ \bar{F}\to(D\wedge E);\\
&4)\ \bar{D}\to \bar{E};\\
&5)\ \overline{(\bar{S}\wedge D)\to E},
\end{aligned}
\qquad (5.16)
$$

где под номером 5 как раз и стоит отрицание формулы (5.14). Далее применяем метод резолюций. Для начала все пять предложений следует представить в виде конъюнкции элементарных дизъюнктов (КНФ). Для простоты дизъюнкцию будем обозначать "+".

Для первого предложения имеем: 1'. $\bar{S}+F$.
Для второго: 2'. $\bar{F}+(\bar{G}\wedge \bar{D}\wedge \bar{E})$.
Для третьего: 3'. $F+(D\wedge E)$.
Для четвертого: 4'. $D+ \bar{E}$.
Для пятого приведем цепочку преобразований:

$$5'.\ \overline{(\bar{S}\wedge D)\to E} = \overline{\overline{(\bar{S}\wedge D)}+E} = \overline{(\bar{\bar{S}}+\bar{D})+E} = \bar{S}\wedge D\wedge E.$$

Пользуясь правилом раскрытия конъюнкции по дизъюнкции, имеем для 2': $(\bar{F}+\bar{G})\wedge(\bar{F}+\bar{D})\wedge(\bar{F}+\bar{E})$;

для 3': $(F+D)\wedge(F+E)$.

Конъюнкция 5' распадается на три одночленных дизъюнкта: S, D, $\bar{E}$.

Теперь у нас имеется система элементарных дизъюнктов, на основе которой проводим вывод методом резолюции.

1. $\bar{S}+F$. Резольвенты
2. $\bar{F}+\bar{G}$. 10. F (6, 9).
3. $\bar{F}+\bar{D}$. 11. $\bar{D}$ (3, 10).
4. $\bar{F}+\bar{E}$. 12. "*Л*" (8, 11).
5. $F+D$. Мы получили "пустой" (ложный)
6. $F+E$. дизъюнкт. Это значит, что выводимость
7. S. формулы (5.14) из системы (5.13) доказана,
8. D. т. е. утверждение (5.13) истинно.
9. $\bar{E}$.

Отметим, что приведенная резолюция не единственно возможная. Можно и по-другому. Например:

10'. $\bar{F}$ (3, 8).
11'. E (6, 10').
12'. "*Л*" (9,11'). И т.п.

Упражнения

1. Пользуясь таблицей истинности, показать равносильность следующих формул:

а. $\bar{b}\wedge(b\to a)$ и $b\to(\bar{b}\wedge a)$;
б. $\bar{r}\to\bar{q}$ и $q\to r$;
в. $(q+\bar{p}\,\bar{q})$ и $((p+\bar{q})(\bar{p}+q))$;
г. $(p+q+r)$ и $(\overline{(a+b)}\to c)$;
д. $(a\to b)$ и $(\overline{(q\wedge\bar{p})})$.

2. Путем простейших преобразований доказать тождества:

а. $((p\to q)\wedge\bar{q})\to\bar{p}$;
б. $(\bar{p}\to\bar{q})\to(q\to p)$;
в. $((p\to q)\to p)\to p$;
г. $(\bar{a}\vee b)\equiv(a\to b)$;

д. $(a\to\bar{b})\equiv(\bar{a}\vee\bar{b})$.

3. Доказать, что формула $f=(a\wedge b)\to a$ тождественна и что подстановка $a=(c\to d)$ не меняет истинности.

а. Те же условия для $f=a\to(a\vee b)$ и $a=c\to(a+b)$.

б. Те же условия для $f=a\to(d\to a)$ и $a=\bar{a}\vee d$.

в. Доказать, что постулат А3 (п.5.5) не меняет тождественности при подстановке $B\to A$ вместо B.

г. То же для постулата А1 (п.5.5) при подстановке $(a+b)\to r$ вместо A.

4. Привести к конъюнктивной нормальной форме.

а. $(p\to q)\to(r\to q)$;
б. $(p\to q)\to((\bar{p}\to q)\to p)$;
в. $(x(y\to z))\to w$;
г. $(\bar{q}\to p)(\bar{q}\to\bar{p})+(p\to r)$;
д. $p\to((p\to q)\to r)$;
е. $(\bar{a}\to b)(c\to d)+(a\to b)$;
ж. $((p\to q)\to(q\to p))+r$;
з. $(\bar{b}(a\to c))\to\bar{a}$;
и. $(p\to q)(\bar{r}\to p)(\bar{q}\to r)+p$;
к. $(a(b\to d))\to c$.

5. Доказать тождества методом редукции.

а. $(\bar{p}\to\bar{q})\to(q\to p)$;
б. $((p\to q)\to p)\to p$;
в. $(p\to q)\to((\bar{p}\to q)\to q)$;
г. Постулат А3 (п.5.5);
д. $(r\to q)\to((p\to q)\to((rp)\to q))$;
е. $A\to(b\to(a+b))$.

6. Принцип дедукции.

Методом опровержения доказать следующие тождества:

а. $(\bar{A}\vee C)\to((\bar{B}\vee C)\to((A\vee B)\to C))$;
б. $(A\to C)\to((B\to C)\to((A\vee B)\to C))$;
в. $(A\to C)\to((B\to D)\to((A\vee B)\to(C\vee D)))$;
г. $(A\to B)\to((A\to\bar{B})\to\bar{A})$;
д. $(A\to C)\to((B\to D)\to((\bar{C}\vee\bar{D})\to(\bar{A}\vee B)))$;

е. $(A \to B) \to ((\overline{A} \to B) \to B)$;

ж. $\overline{(A \wedge B)} \to ((\overline{A} \to C) \to ((\overline{B} \to C) \to C))$;

з. $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$.

7. Пользуясь методом резолюций, доказать следствия:

а. $((a + b), (\overline{a} + c), (\overline{b} + d)) \mapsto (c + d)$;

б. $((a \to b), (a \to c), (\overline{b} + \overline{c})) \mapsto (\overline{a})$;

в. $((a \to c), (b \to d), (\overline{c} + \overline{d})) \mapsto (\overline{a} + \overline{b})$;

г. $((p \to q)(q \to r)) \to (p \to r)$;

д. $(a\,(a \to (b + c))(b \to d)(c \to d)) \to d$,

е. $((\overline{a} \to \overline{b}) \to \overline{(a \to c)}) \to (\overline{a} + \overline{b})$.

8. Записать в виде формул ИВ следующие выражения:

а. Пришел, увидел, победил.

б. Порох состоит из смеси угля, серы и нитрата калия.

в. Волхвы не боятся могучих владык, и княжеский дар им не нужен.

г. Смесь метана и воздуха взрывоопасна.

д. А если надо простой квас медвяный сытить, то взять медку патоки вчетверо да процедить через сито начисто, да положить в сосуд и заквасить простым калачом свежим, и лишь закиснет сливать в бочки.

е. Если руки мыли вы,
если руки мыли мы,
если руки вымыл ты, –
значит, руки вымыты.

ж. Зайку бросила хозяйка,
Под дождем остался зайка,
Со скамейки слезть не мог,
Весь до ниточки промок.

6. ИСЧИСЛЕНИЕ ПРЕДИКАТОВ КАК МЕТОД ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Исчисление высказываний является грубой моделью представления знаний. Основной ее недостаток в том, что высказывание здесь рассматривается как единое целое, без анализа его внутренней структуры. Это ограничивает возможности ИВ при моделировании сложных силлогических построений. Элементарный пример, часто приводящийся в таких случаях. Имеется классический силлогизм:

а. Все люди смертны;

b. Сократ - человек;

c. Следовательно, Сократ смертен.

С точки зрения логики, вывод здесь безупречен, но он уже выходит за рамки ИВ. В самом деле, с помощью пропозициональных связок и букв, его можно записать в виде следующей формулы:

$$(a \wedge b) \to c.$$

Но эта формула необщезначима! А это значит, что логика высказываний не позволяет корректно выразить приведенный силлогизм. Позже мы вернемся к этому примеру и разрешим его, но уже в рамках более гибкой формальной логики, в рамках исчисления предикатов.

6.1. Понятие о предикатах

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

Рассмотрим несколько предложений:

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

В правилах исчисления предикатов эти предложения можно записать следующим образом.

а') СЕСТРЫ (Лена, Таня),
б') В (лес, грибы),
в') ДОЛБИТЬ (капля, камень),
г') БЕЛЫЙ (снег),
д') ПОСЫЛАТЬ (мальчик, брат, книга).

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

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

Термов может быть несколько. По их количеству предикаты разделяются на одноместные (*г'*), двуместные (*а', б', в'*), трехместные (*д'*) и т.д. Предикатная формула еще называется *атомом.* Но и термы бывают разными. В примере *а')* оба терма обозначены вполне конкретно - *Лена, Таня.* В этом случае они называются ***индивидные константы.*** Во втором предикате оба терма заданы в самом общем виде: какие-то грибы в каком-то лесу. Их можно просто обозначить через буквы x и y - они так и называются - ***индивидные*** (***предметные***) ***переменные.*** Сами же предикатные символы, которые, как мы видели, много чего отображают, также обозначаются буквами - прописными, латинского алфавита: P, R, M... Иногда к ним добавляются индексы: P_1, P_2, ... Pn, иногда указывают число мест: $P_к^1$, $P_к^2$... . Говоря о терме, мы не упомянули еще один его вид: терм может быть выражен через функцию.

Разберем все сказанное на примерах.

1) *ПИСАТЬ* (*Лермонтов, "Демон"*). "Лермонтов написал "Демона"" - все ясно: *ПИСАТЬ* - предикатная константа, двуместный предикат, оба терма - индивидные константы. Обозначим через X множество стихотворений Лермонтова. Тогда предикат вида: *ПИСАТЬ* (*Лермонтов,* x) означает: "Лермонтов написал какое-то стихотворение". А вот предикат: *ПИСАТЬ* (y, x), где под y понимается какой-то человек, означает: "кто-то написал что-то". x и y здесь - индивидные переменные.

2) Обозначим через g некоторую функциональную константу, например, "быть варёным". Если картофель обозначить через s, то предикат *НА* (***стол,*** $g(s)$) теперь истолкуется как "на столе варёная картошка". Пусть f - функциональная константа "быть отцом", а m - функциональная константа "быть матерью". В этом случае предикат P (f(*Лена*), m(*Лена*)) следует истолковать просто как *РОДИТЕЛИ*. (Тот же результат, впрочем, даст и более общая формула: $P(f(x), m(x))$, где x - один и тот же ребенок).

Из рассмотренного можно сделать некоторые выводы. Во-первых, термы нельзя менять местами. Иначе получится, что на картошке стоит вареный стол, а Демон написал "Лермонтова". И во-вторых, не следует путать предикатный и функциональный символы. Предикат *МАТЬ* (x,y) означает: y есть мать x. Либо это правда, либо это неправда, поэтому область значений предиката [1,0] или [И,Л]. Функция *м*(x) означает "быть матерью", равенство *м*(x) = y - "матерью x является y". Область определения x - вообще говоря, все человечество, область значений y - все женщины определенного возраста.

6.2. Исчисление предикатов как аксиоматическая система

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

Словарь ИП (A) содержит:

- индивидные константы a, b, c...;
- предметные переменные x, y, z,...;
- функциональные константы f, g, h ..;
- высказывания p, q, r, s,....;
- предикатные константы P, Q, R,.. .

Исчисление предикатов в определенном смысле - продолжение и расширение исчисления высказываний, поэтому в словарь включены все те же пропозициональные связки ∧, ∨, *НЕ*, →, ≡. Но перечень логических знаков в ИП расширяется еще двумя, называемых кванторами: ∀ и ∃. Квантор ∀ читается как "все", "для всех", "всякий", "каков бы ни был" и т.п. Поэтому он называется *квантором всеобщности* (*общности*). Квантор ∃ читается как "некоторый", "хотя бы один", "существует" и т.п.. Поэтому он называется *квантором существования*. Так, например, выражение $\forall xP(x)$ читается: "для любого x выполняется условие $P(x)$". Выражение $\exists yP(y)$ - "существует хотя бы один y, при котором выполняется $P(y)$ (т.е. $P(y) = И$)".

Множество синтаксических правил V ИВ применимо и в ИП. Правильно построенные формулы в рамках исчисления высказываний остаются ппф и в исчислении предикатов. Добавляются правила:

атом есть формула и

если $P(x)$ *формула и* x *- переменная, то* $\forall xP(x)$ *и* $\exists xP(x)$ *формулы.*

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

Рассмотрим формулу $\forall x(R(x,y) \to \exists y(M(x,y,z) \wedge Q(x,y)))$ Здесь все вхождения переменной x связанные, т.к. попадают в область действия квантора $\forall x$, которая включает в себя все предикаты: R,M и Q (следите за скобками). А вот первое вхождение переменной y (в предикате R) - свободное. В дальнейшем y попадает в область квантификации $\exists y$ и является связанным (в предикатах M и Q). Переменная z - свободная.

Каждую предикатную формулу можно интерпретировать, т.е. оценить ее как *И* или *Л*. При этом можно оценить "перекрытие" кванторов на одну и ту же переменную:

$\forall x\exists xP(x)$ интерпретируется как $\exists xP(x)$, а

$\exists x\forall xP(x)$ интерпретируется как $\forall xP(x)$.

Это и понятно: вместо того, чтобы говорить " из всех x существует хотя бы один x, при котором P истинен", достаточно сказать просто: "существует хотя бы один x и т.д.". И наоборот, чтобы не говорить странное словосочетание "существует хотя бы один x, такой что для всех x P истинен", достаточно сказать "для всех x ...". (Для запоминания: из двух кванторов "прав" самый правый).

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

$$\forall x\forall yA = \forall y\forall xA,$$

$$\exists x\exists yA = \exists y\exists xA, \tag{6.1}$$

т.е. одноименные кванторы можно менять местами. Иное дело разноименные кванторы. Здесь выполняется только такое условие:

$$\exists x\forall yA \to \forall y\exists xA. \tag{6.2}$$

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

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

(A4) $\forall xP(x) \to P(y)$,

(A5) $P(y) \to \exists xP(x)$.

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

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

Правила вывода R здесь остаются прежними: правило подстановки и правило заключения, но они дополняются еще одним правилом, учитывающим свойства кванторов. Это правило называется правилом ***специализации***. Суть его в следующем: если ппф $\forall xP(x)$ истинна и b - некоторая константа, то формула *P(b)* также истинна, т.е. справедливо $\forall xP(b) = P(b)$. Пусть, например, имеются формулы $\forall x(P(b) \to Q(x))$ и $P(b)$. Если они истинны, то, применяя специализацию, имеем ряд теорем:

$$\forall x(P(b) \to Q(b)),$$

т.е. $$P(b) \to Q(b), \quad \Big| \quad (\textit{специализация})$$

$$Q(b) \qquad (\textit{modus ponens с } P(b))$$

6.3. Примеры предикатов

Разберем несколько примеров построения предикатов.

1. "А вы, друзья, как ни садитесь, все ж в музыканты не годитесь". Обозначим через x - способ рассаживания музыкантов, y - качество исполнения, $P(x,y)$ - предикат, связывающий способ рассаживания и качество исполнения. Окончательная формула: $\forall x\overline{P}(x,y)$.

2. "Кто не работает, тот не ест":

$$\forall x(P(x) \to E(x)).$$

Здесь x - человек, P - предикатная константа *РАБОТАТЬ*, E - предикатная константа *ЕСТЬ*.

3. "Болтун - находка для шпиона": $\forall x\exists yP(x,y)$, где "роли исполняют": *x - болтун, y - шпион, P - НАХОДКА.*

4. Приведенный в начале главы пример силлогизма о Сократе можно переписать так: для всех x, если x - человек, то x - смертен; Сократ человек; (следовательно) Сократ смертен. Обозначим через M "быть смертным", через H "быть человеком". Мы приходим к следующей формуле:

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

(Сократ здесь - индивидная константа).

5. "В каждом городе найдется краевед, который покажет достопримечательности".

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

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

Здесь к месту отметить некоторые особенности при переводе с живого языка на язык предикатов. Имеется русская фраза "Каждый студент учится". В логической интерпретации это можно отобразить такой формулой:

6. $\forall x(S(x) \to L(x))$,

где *S СТУДЕНТ, L - УЧИТЬСЯ.*

Но вот есть другая фраза: "Некоторые студенты спортсмены". Ее логический эквивалент

7. $\exists x(S(x) \wedge P(x))$ (*P- СПОРТСМЕН*).

Фразы 6 и 7 очень похожи, но замена прилагательного "каждый" на "некоторые" потребовала не только замены квантора $\forall$ на $\exists$, но и замену связки $\to$ на $\wedge$.

И еще. Формулы ИП часто можно писать по-разному на один и тот же словесный текст, пользуясь разной степенью "детализации". Например, короткую фразу из примера 3 можно записать более подробно:

8. $\forall x((M(x) \wedge B(x)) \to \exists yP(x,y))$,

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

А теперь вам предлагается несколько упражнений.

1. Попробуйте представить в виде предикатных формул сле-ующие фразы.

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

2. Пусть L означает ЛЮБИТЬ, *ц* - цветы, *к* - конфеты, д вушка. Переведите на русский язык выражения:

$$a)\ \forall xL(x,ц),$$

$$b)\ \exists x\overline{L}(x,k),$$

$$c)\ \exists xL(x,к),$$

$$d)\ \exists x(L(x,ц) \wedge L(x,к)),$$

$$e)\ \overline{\exists} x(\overline{L}(x,k) \to \overline{L}(x,ц)).$$

. Преобразование формул

Так же, как и в исчислении высказываний, формулы ИП могут ть общезначимыми, выполнимыми (нейтральными) или невы лнимыми (универсально ложными). Вообще говоря, это можно проверить, построив таблицу истинности. Но как это сделать, если предикаты содержат переменные, значения которых в общем случае могут меняться неограниченно? Особенную сложность придает наличие кванторов. Доказать общезначимость при таких условиях совсем не просто. Более того, было показано, что вообще невозможно найти универсального метода установления факта общезначимости квантифицированных выражений, так что даже говорят о неразрешимости исчисления предикатов. Но все это - в общем случае. Практически выполнимость многих формул устанавливать удается, если обозначена область D, в которой связная переменная x принимает свои значения. Если при этом известны истинностные значения формул $A(x)$ и $B(x)$, то легко определяются значения и для формул $\overline{A}$, $A \wedge B$, $A \vee B$, $A \rightarrow B$ и др..

Формула $\forall xA(x)$ получает значение И, если A получает значение И для каждого x из D.

Формула $\exists xA(x)$ истинна, если A истинна хотя бы при одном значении x из D.

Пусть, например, x принимает значения в области D = $(1,2,3)$, и известно, что $A(1)$ = *Л*, $A(2)$ = *И*, $A(3)$ = *Л*. Формула $\forall xA(x)$ означает: для всех x $A(x)$ истинна. Это условие у нас не выполняется, поэтому $\forall xA(x)$=*Л*. Формула $\exists xA(x)$ означает существует x, при котором $A(x)$ истинна. Очевидно, что здесь $\exists xA(x) =$ *И*.

Были найдены процедуры, позволяющие устанавливать общезначимость некоторых ппф. В этом смысле стали говорить, что исчисление предикатов является полуразрешимым. Что же касается свободных переменных, то в большинстве случаев их достаточно *конкретизировать*, т.е. заменить на некоторые постоянные.

При преобразовании формул ИП используются те же приемы: снятие и ограничение отрицания, исключение импликации и эквивалентности, применение законов алгебры логики и правил равносильности и т.п.. При действиях со связками необходимо учитывать особенности квантификаций.

Пусть имеется формула A, не содержащая переменную x. Очевидно, в этом случае она не подпадает под действие кванторов

по x, т.е. $\forall xA = A$ и $\exists xA = A$. Пусть теперь имеется формула B, содержащая свободную переменную x: $B = B(x)$. Тогда справедливы следующие равносильности:

$$\forall x\, B(x) \vee A = \forall x\,(B(x) \vee A),$$
$$\forall x\, B(x) \wedge A = \forall x\,(B(x) \wedge A),$$
$$\exists x\, B(x) \vee A = \exists x\,(B(x) \vee A),$$
$$\exists x\, B(x) \wedge A = \exists x\,(B(x) \wedge A). \qquad (6.3)$$

Если же формулы A и B обе содержат свободную переменную x, то справедливы равенства:

$$\forall x A(x) \wedge \forall x B(x) = \forall x\,(A(x) \wedge B(x)),$$
$$\exists x A(x) \vee \exists x B(x) = \exists x\,(A(x) \vee B(x)). \qquad (6.4)$$

Квантор $\forall$ можно распределять по $\wedge$ (*И*), квантор $\exists$ можно распределять по $\vee$ (*ИЛИ*)).

Дело осложняется, если требуется распределить квантор $\forall$ по *ИЛИ*, а квантор $\exists$ по *И*. В этом случае равносильности, подобные (6.4), не выполняются. Тогда исходят из следующих соображений.

Любая связанная переменная в ппф может принимать какие угодно наименования, так что можно написать, например: $\forall xA(x) = \forall zA(z)$ (то же и для $\exists$). Это действие называется *переименованием переменных*. Всегда стараются сделать так, чтобы каждому квантору соответствовала только одна переменная. Это значительно облегчает работу с кванторами. В дальнейшем мы будем широко этим пользоваться. Вот и в данном случае:

$$\forall xA(x) \vee \forall xB(x) = \forall xA(x) \vee \forall zB(z) = \forall x \forall z(A(x) \vee B(z))$$
$$\exists xA(x) \wedge \exists xB(x) = \exists xA(x) \wedge \exists zB(z) = \exists x \exists z(A(x) \wedge B(z)). \qquad (6.5)$$

(При выносе кванторов за скобки теперь уже использованы отношения (6.3), т.к. $B(z)$, например, не зависит от x и т.п.).

Этот же прием можно использовать и в равенствах (6.4), т.е. справедливо: $\forall xA(x) \wedge \forall xB(x) = \forall x \forall z(A(x) \wedge B(z))$ и т.п..

Здесь и в дальнейшем будем обозначать для удобства:

$$\overline{\forall xP(x)} \Rightarrow \overline{\forall} xP(x) \text{ и } \overline{\exists xP(x)} \Rightarrow \overline{\exists} xP(x).$$

Из общего смысла квантификаций нетрудно понять следующие равенства преобразования отрицаний:

$$\overline{\forall} xP(x) = \exists x\overline{P}(x)\,;\ \overline{\exists} xP(x) = \forall x\overline{P}(x). \qquad (6.6)$$

откуда легко получаются правила замены одного квантора другим:

$$\forall xP(x)=\overline{\exists x\overline{P}(x)}\,;\ \exists xP(x)=\overline{\forall x\overline{P}(x)}\,. \qquad (6.7)$$

В процессе преобразований нам потребуются новые термины. Назовем *литералом* атом или его отрицание; *дизъюнктом* - дизъюнкцию литералов (в ИП это чаще называют *предложением*), *матрицей* - формулу, не содержащую квантификаций.

Если мы имеем матрицу, то дальнейшая работа с ней, в сущности, сводится к преобразованиям, аналогичным тем, которые мы проделывали с формулами исчисления высказываний. Да и цели-то бывают, в общем-то, те же: преобразование и представление в канонических (нормальных) формах, доказательство общезначимости (или невыполнимости), определение логического следствия (чаще всего по методу резолюций). Но для этого нужно научиться получать матрицу, т.е. научиться корректно переходить от формул с квантификациями к формулам, уже не содержащим никаких кванторов. Но сперва еще раз о переименованиях связных переменных, только более подробно.

6.5. Стандартизация переменных

Связные переменные находятся под действием квантора и поэтому переименовывать их можно только одновременно с квантифицированной переменной, стоящей после квантора. Совершенно безразлично, как обозвать квантифицированную переменную. Если у нас есть формула $\forall xP(x)$, то ее можно переобозначить, например, $\forall zP(z)$. Значение истинности ппф при этом не изменится. Такой прием называется переименованием. Им пользуются для разделения переменных с тем, чтобы каждый квантор имел свою, свойственную только ему, переменную. Выше мы уже пользовались этим приемом при выводе соотношений (6.5). Если в формуле $\forall x(P(x)\rightarrow\exists x(R(x,z)))$, провести переименование, то можно получить, например, выражение $\forall x(P(x)\ \rightarrow\exists y(R(y,z)))$ (z просто свободная переменная). Такой прием разделения переменных носит еще название *стандартизации*.

6.6. Исключение квантора существования

Если A не содержит x, то очевидно: $\forall xA\ =\ A$ и $\exists xA\ =\ A$

"Очевидно" - потому что кванторы здесь ничего не определяют и не дают. Это элементарный пример исключения кванторов.

Рассмотрим теперь формулу $\exists xP(x)$ (существует хотя бы один x, при котором $P(x)\ =\ И$). Пусть этот x будет равен c, тогда указанная формула запишется просто $P(c)$. Константа c любая, однако, она не должна совпадать с другими символами, применяемыми в других формулах. Это второй пример исключения квантора существования. Он касается случая, когда сам квантор $\exists$ не находится в области действия какого-либо квантора общности.

Рассмотрим теперь формулу $\forall x\exists yP(x,y)$ (для всех x существует по крайней мере один y такой, что выполняется предикат $P(x,y)$). Ясно, что y, удовлетворяющий этому условию, как-то зависит от x. Эту зависимость можно отобразить с помощью некоторой функции, например, $g(x)$. Эта функция теперь заменяет y, а квантор существования можно просто убрать:

$$\forall xP(x,g(x)).$$

Функция типа $g(x)$, отображающая каждое значение x в "тот самый y", называется функцией Сколема или сколемовской.

Особенности:

1)если квантор $\exists$ стоит перед квантором общности $\forall$, то он не находится в области его действия и поэтому заменяется, как и в прежнем примере, некоторой константой:

$$\exists x\forall yP(x,y)\ =\ \forall yP(a,y).$$

2)если квантор $\exists$ находится в области действия нескольких кванторов общности, то соответствующая его переменная заменяется сколемовской функцией от соответствующего числа переменных (мест):

$$\forall x\forall y\forall r\,\exists wP(x,y,r,w)\ =\ \forall x\forall y\forall rP(x,y,r,g(x,y,r))$$

"собственная" переменная квантора существования и заменяется сколемовской функцией g, зависящей от всех квантифицированных переменных, в сфере действия которых она находится

3)Следующий пример говорит сам за себя:

$$\exists x\exists yP(x,y)\ =\ P(a,b).$$

4)Еще один пример:

$$\forall x[P(x,y)\wedge\exists y(M(y,z)\vee R(y,z,q))]=\forall x[(P(x,y)\wedge(M(g(x),z)\vee R(g(x),z,q))]$$

Сколемовская функция $g(x)$ заменяет y во всех местах его вхож дения в области действия квантора ∃.

Рассмотрим и осмыслим теперь такой обобщающий пример:

5)Дано: $\exists z\forall x\exists u\forall y\forall r\exists w\exists s\forall v\exists qM(z,x,u,y,r,w,s,v,q)$.

После сколемизации:

$$\forall x\forall y\forall r\forall vM(a,x,f(x),y,r,g(x,y,r),h(x,y,r),v,p(x,y,r,v)).$$

6.7. Предваренная форма

Полученная только что, после сколемизации формула имеет вполне определенный вид. Она состоит из цепочки кванторов, называемой *префиксом* и бескванторной формулы, называемой *матрицей*. Представить какую-либо формулу в виде префикса и матрицы - это значит представить ее в *предвáренной форме*. Особенность предваренной формы в том, что все кванторы оказываются вынесенными влево за пределы общей формулы, а часть, оставшаяся без кванторов, может быть подвергнута всем возможным преобразованиям, в частности, быть представленной в конъюнктивной нормальной форме, такой необходимой нам для реализации принципа резолюции.

Для любой логической формулы существует логически эквивалентная ей предваренная форма.

Это правило вытекает из тех преобразований, которые необходимо для этого проделать, - они известны и всегда выполнимы:

- исключить связки эквивалентности и импликации;
- переименовать (если необходимо) связанные переменные таким образом, чтобы каждый квантор имел свою переменную;
- удалить те квантификаторы, область действия которых не содержит квантифицированной переменной, как ненужные;
- ограничить область действия отрицания;
- провести сколемизацию;
- переместить все кванторы общности в начало формулы, образовав префикс и матрицу.

К моменту последней операции перемещения кванторов связанные переменные уже разделены, каждый квантор общности имеет свою переменную, независимые переменные заменены постоянными (конкретизация). Будучи перемещенным в начало формулы, они все равно распространяют влияние лишь на "свои" переменные. Широко используются приведенные ранее равносильности (6.1) - (6.7). Может пригодиться еще и правило раскрытия импликации:

$$\exists x(A(x)\rightarrow B(x)) = \forall xA(x)\rightarrow \exists xB(x). \qquad (6.8)$$

6.8. Исключение кванторов общности

Получение предваренной формы - важный этап в деле преобразования исходной формулы ИП. Дальнейшая работа будет проводиться с матрицей, все переменные которой относятся к тому или иному квантору общности. Роль цепочки кванторов (префикса) в данном случае - в простом напоминании об этом факте. К тому же порядок кванторов общности роли не играет. Если все это принять во внимание, то префикс вообще можно убрать и предположить по соглашению, что все переменные относятся к своим кванторам общности. Остается одна матрица.

6.9. Приведение матрицы к КНФ

Если теперь рассматривать матрицу как логическое выражение, связывающее посредством пропозициональных связок некоторое количество литералов, то процесс приведения любой матрицы к конъюнктивной нормальной форме будет мало чем отличаться от такого же процесса в рамках исчисления высказываний. Процесс этот всегда возможен, и конечный итог известен: матрица будет представлена в виде конъюнкции дизъюнктов (каузальная форма). Дизъюнкт, напомним, в исчислении предикатов называется предложением. Знаки конъюнкции можно опустить, и все множество полученных предложений представить в виде столбца, рассматривая каждое из них как некую исходную гипотезу (как это мы делали с дизъюнктами в процессе логического вывода в ИВ).

6.10. Обобщающий пример

Привести заданное выражение к системе предложений.

$$\forall x\{\forall y[P\,(x,y)\rightarrow \exists xQ(x,\,h(y))]\rightarrow \forall x\exists yR(a,y,x)\}.$$

Освобождаемся от импликаций.

$$\forall x\{\overline{\forall} y[\overline{P}\,(x,y)\vee \exists xQ(x,h(y))]\vee \forall x\exists yR(a,y,x)\}.$$

Переносим и снимаем отрицания (согласно (6.6) и др. правилам).

$$\forall x\{\exists y[\overline{\overline{\overline{P}}(x,y)\vee\exists xQ(x,h(y))]}\vee\forall x\exists yR(a,y,x)\}$$

$$\forall x\{\exists y\,[\overline{\overline{P}}(x,y)\wedge\overline{\exists}xQ(x,h(y))]\vee\forall x\exists yR(a,y,x)\}\,.$$

Окончательно:

$$\forall x\{\exists y\,[P(x,y)\wedge\exists x\overline{Q}(x,h(y))]\vee\forall x\exists yR(a,y,x)\}\,.$$

Проводим стандартизацию: каждый квантор должен иметь свою переменную. С этой целью переименуем в средней формулс (для $\overline{Q}$) x на u, в последней формуле - x на z, y на w:

$$\forall x\{\exists y\,[P(x,y)\wedge\forall u\overline{Q}(u,h(y))]\vee\forall z\exists wR(a,w,z)\}.$$

Исключаем кванторы существования. Заметим, что квантор $\exists y$ находится в сфере действия $\forall x$ и распространяется на квадратные скобки, а квантор $\exists w$ - в сфере действия $\forall x$ и $\forall z$, соответственно вводим функции Сколема.

$$\forall x\{\,[P(x,f(x))\wedge\forall u\overline{Q}(u,h(f(x)))]\vee\forall zR(a,g(x,z),z)\}.$$

Выносим кванторы общности влево, за скобки, образуем предваренную форму.

$$\forall x\forall u\forall z\;\{\,[P(x,f(x))\wedge\overline{Q}(u,h(f(x)))]\vee R(a,g(x,z),z)\}.$$

Опускаем кванторы общности, полученную матрицу приводим к КНФ.

$$[P(x,f(x))\wedge\overline{Q}(u,h(f(x)))]\vee R(a,g(x,z),z)=$$

(раскрывая конъюнкцию по дизъюнкции)

$$=(P(x,f(x))\vee R(a,g(x,z),z))\wedge(\overline{Q}\,(u,h(f(x)))\vee R(a,g(x,z),z)).$$

Окончательно получаем систему предложений:

1. $P(x,f(x))\vee R(a,g(x,z),z))$,
2. $\overline{Q}(u,h(f(x)))\vee R(a,g(x,z),z))$.

Хорошо, что мы научились приводить выражение в предваренной форме к системе предложений. Но для того, чтобы идти дальше и проводить с ними процесс резолюции, нам необходимо уметь находить контрарные литералы, что при участии многоместных предикатов с различными переменными становится делом весьма непростым. Пусть, например, имеются предложения:

1. $Q\,(u)\vee P\,(A)$,
2. $\overline{Q}(w)\vee P\,(w)$,
3. $\overline{Q}(x)\vee\;\overline{P}\,(x)$
4. $Q\,(y)\vee\;\overline{P}\,(y)$.

Судя по предикатным символам, здесь вполне возможна резолюция. Но различные переменные не дают нам делать это непосредственно. Необходимо делать *подстановки*.

6.11. Подстановки и унификация

Пусть имеется формула $P(x,y,z)$. Ввести подстановку s это значит, определить множество пар типа $s=\{t_1/x,\ t_2/y,\ t_3/z\}$,позволяющих заменить x на t_1, y на t_2, z на t_3 во всех местах их вхождений. Полученная формула равносильна прежней: $P(x,y,z)=P(t_1,\ t_2,\ t_3)$.

Подстановки открывают широкие возможности при преобразовании формул. Простейший пример:

дана формула $P(x,f(y),z)\wedge P(a,b,r)$.

Введем подстановку $s_1=\{a/x,b/f(y),r/z\}$ для первого предиката и получим: $P(a,b,r)\wedge P(a,b,r)=P(a,b,r)$, т.е. произошла "склейка" - распространенный прием в процессах вывода.

Проводя подстановки, надо следовать следующим правилам:

- подстановки применяются только для свободных переменных, в том числе и для функциональных, во всех местах их вхождений в данную формулу (предложение);
- переменную можно заменить константой, но нельзя константу заменить переменной;
- переменная не может быть заменена на терм, содержащий ту же самую переменную;
- подстановки могут касаться только некоторых термов предиката, остальные термы остаются без изменения;
- одна функция не может подставляться вместо другой, но может заменять переменную-аргумент;
- нельзя вместо свободной переменной ввести уже связанную, другими словами, вводимое подстановкой вхождение переменной не должно попадать в область действия квантора, связывающего данную переменную.

К последнему правилу приведем простой пример. Формула $\exists y(x<y)$ отображает вполне корректное математическое условие (для любого x всегда найдется y, что $x<y$). Некорректная подстановка y/x приводит к ложному суждению: $\exists y\,(y<y)$.

Если к формуле E применена подстановка s_1, то пишут Es_1.

Так, если к предикату $P(x,f(y),B)$ применена подстановка $s_1 = \{z/x,w/y\}$, то можно написать $P(x,f(y),B)s_1 = P(z,f(w),B)$.

Если к полученной формуле применить еще подстановку $s_2=\{C/z, D/f(w)\}$, то получится $P(x,f(y),B)s_1s_2 = P(C,D,B)$.

Если все термы предиката не содержат переменных, то мы имеем *"основной частный случай" ("основной пример")*.

Последовательное применение подстановок называется их композицией.

Некоторая подстановка s называется *унифицированной* (унификатором), если ее применение к ряду формул E_1,E_2, ... делает их равными: $E_1s = E_2s = E_3s$ ... Например, подстановка $s = \{A/x,B/y\}$ унифицирует формулы $P(x,f(y),B)$ и $P(x,f(B),B)$ и дает $P(A,f(B),B)$.

Примечание. Обратите внимание, что предикатные символы при этом должны быть одинаковыми.

Освоить технику унификации совершенно необходимо. В процессе резолюции, например, контрарные литералы должны быть идентичны, что как раз и достигается их унификацией. Разберем поэтому несколько показательных примеров.

Пример 1. Унифицировать множество литералов:

$$\{P(a,x), P(y,f(b))\}.$$

Поиск возможных подстановок начинаем с установления различий в термах обоих литералов, начиная с левых позиций. Первое рассогласование: (a,y). Так как переменной здесь является y, то вводим подстановку $s_1= (a/y)$. Получаем пару литералов

$$\{P(a,x), P(a,f(b))\}.$$

Очередное рассогласование: $(x,f(b))$. Теперь уже вводим подстановку $s_2 = (f(b)/x)$: $\{P(a,f(b)), P(a,f(b))\}$. Унификация состоялась. Общая подстановка имеет вид $su = s_1s_2 = (a/y, f(b)/x)$.

Подстановка su, определенная таким образом, называется ***наиболее общим унификатором - ноу***.

Пример 2. Определить *ноу* для пары литералов

$$P(x,f(x),z) \text{ и } P(A,u,u).$$

Начинаем с левых позиций. Рассогласование: (x,A), подстановка $s_1 = (A/x)$, литералы: $P(A,f(A),z)$ и $P(A,u,u)$. Рассогласование $(f(A),u)$, $s_2 = (f(A)/u)$, литералы: $P(A,f(A),z)$ и $P(A,f(A),f(A))$. Рассогласование: $(z,f(A))$, $s_3 = (f(A)/z)$.

Окончательно:

$$P(A,f(A),f(A)) \text{ и } P(A,f(A),f(A)).$$

Наиболее общий унификатор $su = s_1s_2s_3 = (A/x,f(A)/u,f(A)/z)$.

6.12. Вывод в исчислении предикатов

Так же, как и в исчислении высказываний, проблема вывода в ИП сводится к проблеме дедукции, т.е. к решению вопроса: является ли формула B логическим следствием множества формул $\{E\}$. Напомним, что ппф B является логическим следствием множества $\{E\}$, если она принимает значение *И* всякий раз, как только все Ei одновременно принимают значение *И*. Вопрос решался через доказательство теоремы дедукции, которая справедлива и в ИП, с учетом, впрочем, ряда оговорок и условий, касающихся действий со связными и свободными переменными. Так же, как и в ИВ, встает проблема определения общезначимости той или иной формулы. И решается она примерно так же, т.е. как минимум пятью способами, каждый из которых применим в ИП:

1) оценка с помощью таблицы истинности,

2) оценка ~~через~~ преобразование, упрощение и приведение к нормальным формам,

3) оценка путем логического вывода из системы аксиом,

4) оценка методом редукции,

5) оценка методом опровержения.

Однако самым эффективным средством порождения и/или доказательства следствий здесь по-прежнему является метод резолюций.

В ИП резолюция применяется к предложениям, представляющим из себя дизъюнкцию литералов, каждый из которых, в общем случае, содержит различные переменные. Очевидно, что родительские предложения должны содержать контрарные литералы, идентичные по форме. Идентичность литералов достигается при этом применением соответствующих подстановок и унификации.

Пусть, например, у нас имеется два родительских предложения L и M, содержащие соответственно литералы l и $\overline{m}$, которые допускают общую подстановку-унификатор s такую, что $(l)s$

= *(m)s* (без учета инверсии). Теперь возможна резолюция, которая "уничтожит" оба этих литерала, но при этом надо помнить, что подстановка возможна только для предложения в целом. Поэтому результат резолюции можно записать в виде нового предложения, объединяющего (*U*) оставшиеся литералы:

$$(L-l)\,s\bigcup(M-m)\,s. \qquad (6.9)$$

Родительских предложений может быть несколько, так же как и пар контрарных литералов. Резольвенту (6.9) перепишем теперь в общем виде:

$$(L-\{l_i\})\,s\bigcup(M-\{m_i\})\,s. \qquad (6.10)$$

Рассмотрим следующий пример. Имеются два предложения:

$$L\colon P\,[x,f(A)]\vee P\,[x,f(y)]\vee Q(y),$$
$$M\colon \overline{P}\,[z,f(A)]\vee\overline{Q}\,(z).$$

Для резолюции выбираем литералы: $l=P\;[x,f(A)]$ и $m=\overline{P}\,[z,f(A)]$.

Подстановка $s=z/x$ для них является унификатором (применяя ее к l). Но при этом все предложение L примет вид:

$$La\colon P\,[z,f(A)]\vee P\,[z,f(y)]\vee Q(y).$$

Результат резолюции *La* и *M* согласно (6.9):

$$P[z,f(y)]\vee Q(y)\vee\overline{Q}\,(z).$$

Совершенно иной результат получится, если выбрать множество $\{l_i\}=(P\,[x,f(A)],\ P\,[x,f(y)])$, а $m=\overline{P}\,[z,f(A)]$.

Введем унификатор $s=(z/x,\ A/y)$. Исходные предложения принимают вид:

$$Lb\colon P\,[z,f(A)]\vee P\,[z,f(A)]\vee Q(A),$$
$$M\colon \overline{P}\,[z,f(A)]\vee\overline{Q}\,(z).$$

Два первых литерала предложения *Lb* склеиваются в один, и в результате получится резольвента $Q\,(A)\vee\overline{Q}\,(z)$.

Здесь возможны и другие резолюции, в частности, по *Q*.

Примечание. Термы в многоместном предикате не могут меняться местами, поэтому литералы $P\,(x,f(y))$ и $P\,(f(y),x)$ не идентичны. Это следует учитывать в процессе резолюции.

Так же, как и в исчислении высказываний, метод резолюций основной инструмент доказательства логического следствия по методу опровержения. Принцип дедукции, который при этом реализуется, все тот же: если ппф *B* является логическим следствием системы $\{E\}$, то справедливо $\{E,\overline{B}\}\mapsto Л$, т.е. множество $\{E,\overline{B}\}$ имеет своим следствием пустой дизъюнкт. Надо только помнить, что все формулы здесь - ппф в смысле исчисления предикатов со всеми вытекающими отсюда действиями и ограничениями.

Пусть заданы две гипотезы:

1. $\forall x[P(x)\vee\exists xQ(x)\vee R(x)]$,
2. $\overline{\forall}yQ(y)$.

Требуется доказать, что выражение

3. $\overline{\exists}zP(z)\to\forall zR(z)$

является логическим следствием этих двух. Следуя принципу дедукции, находим отрицание выражения 3:

$$\overline{(\overline{\exists}zP\;(z)\to\;\forall zR\;(z))}.$$

Исключая импликацию и приводя в порядок отрицания, получаем:

$$\overline{(\overline{\overline{\exists}}zP\,(z)\vee\forall zR\,(z))}=\overline{(\exists zP\,(z)\vee\forall zR\,(z))}=$$
$$\overline{\exists}zP(z)\wedge\overline{\forall}zR(z)=\forall z\overline{P(z)}\wedge\exists z\overline{R(z)}$$

после переименования и сколемизации – к предваренной форме),

$$\forall z\overline{P(z)}\wedge\exists u\overline{R(u)}=\forall z\overline{P(z)}\wedge\overline{R(b)}.$$

Или окончательно:

4. $\forall z\left(\overline{P(z)}\wedge\overline{R(b)}\right)$.

Гипотеза 1 преобразуется просто (вводится функция Сколема):

5. $\forall x[P(x)\vee\exists wQ(w)\vee R(x)]=\forall x[P(x)\vee Q(g(x))\vee R(x)]$,

для гипотезы 2 используем равенство:

6. $\overline{\forall}yQ(y)=\exists y\overline{Q}(y)=\overline{Q}(a)$.

Кванторы общности в предваренных формах выражений 4 и теперь можно опустить. Мы приходим к системе предложений

а) $P(x)\vee Q(g(x))\vee R(x)$,

б) $\overline{Q}(a)$,

в) $\overline{P}(z)$,

г) $\overline{R}(b)$.

Знак $\wedge$ в матрице выражения 4 опускается, и она распадается на два предложения в) и г)).

Теперь можно проводить резолюции. У каждой резольвенты будем указывать унификатор, позволяющий ее получить.

д) $Q(g(z))\vee R(z)$, $s = z/x$ (а, в)

$Q(g(B))$, $s = B/z$ (г, д),

ж) *Л*, $s = A/g(B)$ (б, е).

Логическое следствие доказано.

6.13. Примеры применения метода резолюций

Пример 1. Ранее нами был рассмотрен силлогизм о Сократе и получена соответствующая предикатная формула:

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

(Здесь *H* - быть человеком, *M* - быть смертным). Требуется доказать, что эта формула общезначима. Докажем это методом опровержения: формула общезначима, если ее отрицание невыполнимо. Берем отрицание исходной формулы (*Сократ* обозначим *C*).

$$\overline{(\forall x((H(x)\to M(x))\wedge H(C))\to M(C))}.$$

Исключаем импликации, работаем с отрицаниями.

$$\overline{(\overline{\forall x((\overline{H}(x)\vee M(x))\wedge H(C))}\vee M(C))},$$

$$\overline{(\exists x\overline{((\overline{H}(x)\vee M(x))\wedge H(C))}\vee M(C))},$$

$$\overline{(\exists x(\overline{(\overline{H}(x)\vee M(x))}\vee\overline{H(C)})\vee M(C))},$$

$$\overline{(\exists x((H(x)\wedge\overline{M}(x))\vee\overline{H(C)})\vee M(C))},$$

$$\forall x(\overline{(H(x)\wedge\overline{M(x)})\vee\overline{H(C)}})\wedge\overline{M(C)},$$

$$\forall x(\overline{(H(x)\wedge\overline{M(x)})}\wedge H(C))\wedge\overline{M(C)},$$

$$\forall x((\overline{H(x)}\vee M(x))\wedge H(C))\wedge\overline{M(C)}.$$

Мы получили предваренную форму в конъюнкции с $\overline{M(C)}$.

Опуская квантор общности и ненужные скобки, получаем выражение, уже представленное в КНФ:

$$(\overline{H}(x)\vee M(x))\wedge H(C)\wedge\overline{M}(C).$$

Таким образом, получается следующая система предложений

1. $\overline{H}(x)\vee M(x)$,

2. $H(C)$,

3. $\overline{M}(C)$.

Применяя подстановку $s=C/x$, проводим резолюцию.

4. $M(C)$, (1,2)

5. *Л*. (3,4)

Общезначимость исходного силлогизма доказана.

Пример 2. Посылка: каждый предприниматель, который занимается бизнесом, получает прибыль. Заключение: если прибыли нет, никто заниматься бизнесом не будет. Введем обозначение предикатных констант: $S(x,y)$ - *x ЗАНИМАЕТСЯ ДЕЛОМ y*; *B* - *БИЗНЕС*; *P* - *ПРИБЫЛЬ*; $R(x,y)$ - *x ПОЛУЧАЕТ y*. Посылка запишется в следующем виде:

$$\forall x(\exists y(S(x,y)\wedge B(y))\to\exists y(P(y)\wedge R(x,y)))$$

(если у каждого предпринимателя *x* есть дело *y И* это дело - бизнес *B*, то это дело приносит прибыль *P, И* он ее получает (*R*)), а заключение - в виде:

$$\overline{\exists}xP(x)\to\forall x\forall y(S(x,y)\to\overline{B}(y))$$

(если никто не получает прибыль, то каким бы делом он ни занимался, это будет не бизнес).

Представим посылку в виде системы предложений. Исключаем импликацию и делаем преобразования.

$$\forall x(\overline{\exists}y(S(x,y)\wedge B(y))\vee\exists y(P(y)\wedge R(x,y))),$$

$$\forall x(\forall y(\overline{S(x,y)\wedge B(y)})\vee\exists y(P(y)\wedge R(x,y))),$$

$$\forall x(\forall y(\overline{S}(x,y)\vee\overline{B}(y))\vee\exists y(P(y)\wedge R(x,y))).$$

После сколемизации – к предваренной форме:

$$\forall x(\forall y(\overline{S}(x,y)\vee\overline{B}(y))\vee\exists w(P(w)\wedge R(x,w))),$$

$$\forall x(\forall y(\overline{S}(x,y)\vee\overline{B}(y))\vee(P(f(x))\wedge R(x,f(x))))$$

(заметим, что квантор $\exists w$ (после переименования) зависит только от $\forall x$). Матрицу приводим к КНФ.

$$(\overline{S}(x,y)\vee\overline{B}(y))\vee(P(f(x))\wedge R(x,f(x))$$

и, раскрывая конъюнкцию, получаем предложения посылки.

1. $\overline{S}(x,y)\vee\overline{B}(y)\vee P(f(x))$,

2. $\overline{S}(x,y)\vee\overline{B}(y)\vee R(x,f(x))$.

Теперь займемся заключением. Надо найти его отрицание.

$$\overline{(\overline{\exists}xP(x) \to \forall x\forall y(S(x,y) \to \overline{B}(y)))}\ ,$$

$$\overline{(\overline{\overline{\exists}}xP(x) \vee \forall x\forall y(\overline{S}(x,y) \vee \overline{B}(y)))}\ ,$$

$$\overline{(\exists xP(x) \vee \forall x\forall y(\overline{S}(x,y) \vee \overline{B}(y)))}\ ,$$

$$\overline{\exists}xP(x) \wedge \overline{\forall}x\forall y(\overline{S}(x,y) \vee \overline{B}(y))\ ,$$

$$\forall x\overline{P}(x) \wedge \exists x\exists y\overline{(\overline{S}(x,y) \vee \overline{B}(y))}\ ,$$

$$\forall z\overline{P}(z) \wedge \exists x\exists y(S(x,y) \wedge B(y)),$$

(переименование и сколемизация)

$$\forall z\overline{P}(z) \wedge (S(a,b) \wedge B(b)).$$

Матрица: $\overline{P}(z) \wedge \mathrm{S(a,b)} \wedge \mathrm{B(b)}$. Предложения от заключения:

3. $\overline{P}(z)$, 4. $S(a,b)$, 5. $B(b)$.

Резолюции для всех предложений 1-5:

6. $\overline{S}(x,y) \vee \overline{B}(y)$ (1,3) ($s=f(x)/z$),
7. $B(b)$ (4.6) ($s=a/x,\ b/y$),
8. $Л$ (5,7).

Наконец, рассмотрим еще один пример, который пригодится нам в дальнейшем.

Пример 3. Имеются следующие утверждения:

Кто работает с интегралами, тот математик

1. $\forall x[\,И(x) \to M(x)\,]$.

Дети не математики

2. $\forall x[\,Д(x) \to \overline{M}(x)\,]$.

Некоторые дети обладают математическими способностями

3. $\exists x\,[\,Д(x) \wedge C(x)\,]$.

Требуется доказать следующее заключение: некоторые из тех, кто имеет способности к математике, не работают с интегралами:

4. $\exists x[C(x) \wedge \overline{И}(x)]$.

Первое и второе утверждения приводятся к предложениям:

1'. $И(x) \vee M(x)$,

2' $Д(x) \vee \overline{M}(x)$.

В третьем проведем сколемизацию, оно распадается на два однолитерных предложения:

3'. $Д(a)$;

4'. $C(a)$.

От четвертого надо взять отрицание.

$$\overline{\exists}x[C(x) \wedge \overline{И}(x)]\ ,$$

$$\forall \overline{x[C(x) \wedge \overline{И}(x)]}\ ,$$

$$\forall x[\overline{C}(x) \vee \overline{\overline{И}}(x)]\ .$$

Получаем последнее предложение

5. $\overline{C}(x) \vee И(x)$.

С предложениями 1'- 4' и 5 проводим резолюцию (соответствующие подстановки подразумеваются).

6. $И(a)$ (4',5),
7. $M(a)$ (6,1'),
8. $\overline{Д}(a)$ (7,2'),
9. $Л$ (8,3').

6.14. Стратегии резолюции

Речь идет о применении метода резолюций при реализации опровержения. Если резолюции нужны нам только лишь для порождения возможно большего количества следствий, то это вопрос не слишком острый. Ибо именно порождение большого количества резольвент является свойством и особенностью применения резолюций. Но тут возникает вопрос о выборе эффективной стратегии поиска результата, вопрос, пока еще не нашедший однозначного решения.

Метод полного перебора - самый простой по идее и по алгоритму. Еще он называется методом поиска в ширину. Знакомое название. А удивляться нечему: множество резольвент – это ведь математическая модель, отображающая множество состояний предметной области. Поиск в пространстве состояний в определенном смысле, аналогичен поиску решения на множестве резольвент.

Начинается он с поиска резольвент между предложениями базового множества. (Базовым будем называть множество предложений посылок вкупе с предложениями, полученными при отрицании заключения). Это будут резольвенты первого уровня. По мере их вычисления они подсоединяются к концу списка предложений. Далее вычисляются резольвенты 2-го уровня как результаты резолюций базового множества дизъюнктов и дизъюнктов 1-го уровня или их потомков и т.д. Метод можно проследить на простом примере. Пусть S заданное множество (4 дизъюнкта).

S:	Резольвенты:			
	1-й уровень		2-й уровень	
1. $P \vee Q$	5. Q	(1,2)	13. $P \vee Q$	(1,7)
2. $P \vee Q$	6. P	(1,3)	14. $P \vee Q$	(1,8)
3. $P \vee Q$	7. $Q \vee Q$	(1,4)	15. $P \vee Q$	(1,9)
4. $P \vee Q$	8. $P \vee P$	(1,4)	16. $P \vee Q$	(1,10)
	9. $Q \vee Q$	(2,3)	17. Q	(1,11)
	10. $P \vee P$	(2,3)	18. P	(1,12)
	11. P	(2,4)	......	
	12. Q	(3,4)	и т.д.	

Этот бесхитростный перебор не сулит быстрого успеха, но зато гарантирует нахождение пустого дизъюнкта (если он, конечно, имеется). Такая стратегия называется *полной*. По указанной стратегии дизъюнкт ***Л*** будет 39-м!

Из примера следуют важные выводы. Так, оказалось, что было порождено много ненужных и потому лишних дизъюнктов. Многие из них повторяются: 5 и 17, 6 и 18, 13-16. Некоторые являются тавтологиями: 7-10. При поиске по методу опровержения мы ищем невыполнимый дизъюнкт и эту невыполнимость нам нельзя потерять. Тавтологии же всегда выполнимы, к делу не относятся и могут быть вычеркнуты, иначе они сами начнут порождать лишние резольвенты. Если из списка порождаемых дизъюнктов исключать повторяющиеся, а также тавтологии, то число резольвент значительно уменьшится и метод полного перебора окажется не таким уж громоздким. Его можно еще более оптимизировать, если применить так называемый *принцип подсуммирования (поглощения)*.

Пусть у нас имеются два дизъюнкта C и D, такие, что $C = A$, $D = A \vee B$. C, как видим, является составной частью D. Говорят, что *C подсуммирует D*. Если теперь окажется, что D невыполним (=*Л*), то по свойству дизъюнкции $C = A =$ *Л*. Замена D на C ***сохраняет невыполнимость***, и D можно просто отбросить: *C поглощает D*.

Примеры подсуммирования:

C:		D:
$A \vee R$	подсуммирует	$A \vee B \vee R$,
$P(x)$	подсуммирует	$P(y) \vee Q(z)$,
$P(x) \vee Q(E)$	подсуммирует	$P(f(A)) \vee Q(B) \vee R(z)$,
$P(x) \vee Q(f(x),a)$	подсуммирует	$P(q(y)) \vee Q(f(q(y)),a) \vee R(z)$.

Обратите внимание, что подсуммирование возможно после введения соответствующих подстановок. Только после этого допустимо поглощение и правые дизъюнкты отбрасываются в пользу левых.

Вычеркивая ненужные дизъюнкты и применяя подсуммирование, мы в определенном смысле видоизменяем сам метод полного перебора. Мы применяем теперь *стратегию упрощения*. Обладая полнотой исходного метода, такая стратегия существенно сокращает список резольвент. Алгоритм перебора остается прежним, но каждая возникающая резольвента анализируется и при необходимости отбрасывается. Так, рассмотренный выше пример разрешится следующим образом:

Резолюции:	5. Q	(1,2);	6. P	(1,3);
	7. $\overline{P}$	(2,4);	8. $\overline{Q}$	(3,4);
	9. Л	(5,8) – (начало второго уровня).		

Дерево опровержения. Выбранную стратегию опровержения удобно отображать в виде графа с корневой вершиной с пометкой *Л*. Такой граф не только обладает наглядностью, но и предостерегает от ошибочных или просто повторных шагов. Выше мы рассматривали пример с математиками (пп.6.13). Построим соответствующее ему дерево опровержения. Для каждого базового предложения отведем свою вершину. Если две вершины образуют родительскую пару, то ребра, исходящие из них, будут сходиться к вершине, помеченной их резольвентой-потомком.

Приведенный пример удобен тем, что он иллюстрирует сразу несколько стратегий.

Стратегия опорного множества. Согласно этой стратегии процесс опровержения строится так, что по крайней мере один из родителей в каждой резольвенте выбирается из предложений, полученных при отрицании заключения (целевая ппф) или их потомков (опорное множество). Эта стратегия обладает полнотой, т.е. можно показать, что если опровержение существует, оно может быть найдено по алгоритму стратегии опорного множества. Его применение приводит к более медленному росту числа резольвент, особенно в ширину, при большей интенсивности проникновения в глубину (увеличивая количество уровней). Такое движение как раз способствует поиску пустого дизъюнкта. Дерево опровержения на рис.6.1 могло бы быть порождено стратегией опорного множества.

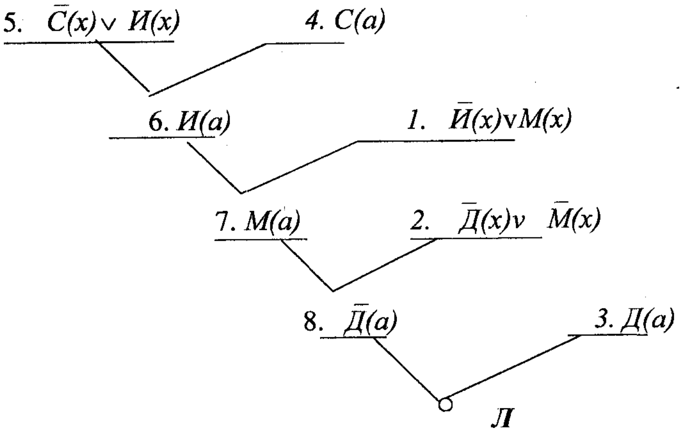


Рис. 6.1. Дерево опровержения

Стратегия предпочтения одночленам. Можно сказать, что эта стратегия есть модификация предыдущей. Стратегия эта заключается в том, что на каждом шаге резолюции мы пытаемся в качестве одной из родительских вершин выбирать однолитерное предложение. В такой стратегии заключается определенный смысл: когда в резолюции используется одночлен, резольвента содержит

меньшее по сравнению с другим родителем число литералов. Процесс целенаправленно движется в сторону увеличения однолитерных резольвент, что способствует получению пустого дизъюнкта. Рис. 6.1. может служить примером указанной стратегии.

Стратегия, линейная по входу. Особенность линейной стратегии в том, что в каждой резолюции одно из родительских предложений принадлежит базовому множеству. Такое начало нам уже знакомо по стратегии полного перебора. Но там, повинуясь другой идее, алгоритм постепенно уходит в сторону. Если же следовать стратегии, линейной по входу, то можно было бы заметить, что она приводит к заметному снижению порождаемых дизъюнктов. Отметим, что граф опровержения на рис.6.1 соответствует стратегии, линейной по входу. (Обратите внимание, что в данном случае на каждом шаге резолюции здесь участвует резольвента, полученная на предыдущем шаге. Такое дерево называется дерево-лоза). Заметим, однако, что известны случаи, когда эта стратегия не приводила к желаемому результату, хотя опровержение заведомо существовало. Это означает, что стратегия, линейная по входу, не является полной, что следует учитывать при ее использовании. Так, мы сейчас рассмотрим простой пример, когда строгое следование лозе не приводит к желаемому и такому близкому результату, но небольшое изменение в алгоритме дает прекрасный результат.

Пример. Имеется система предложений:

$$R(x) \vee \bar{P}(x),$$
$$\bar{R}(y) \vee \bar{P}(y),$$
$$\bar{R}(z) \vee P(z),$$
$$R(h) \vee P(A).$$

Соответствующее ей дерево-лоза приводится на рис.6.2. Необходимые подстановки здесь очевидны.

Среди базовых предложений не нашлось однолитерного для образования пустого дизъюнкта. Резолюция с одним из потомков спасла положение.

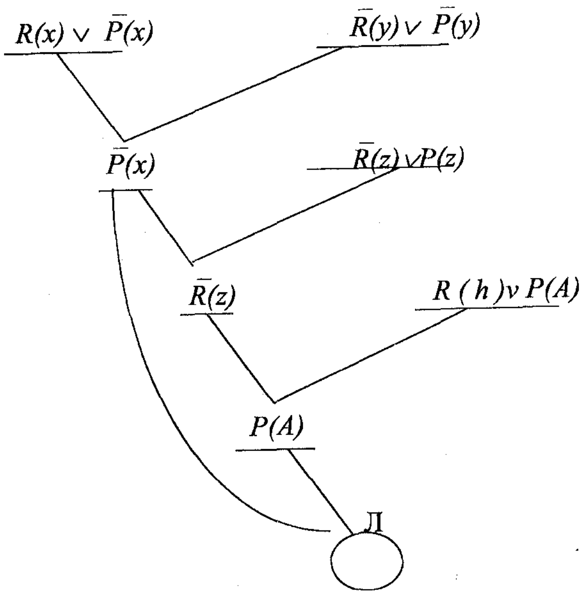


Рис 6.2. Опровержение, линейное по входу, с коррекцией алгоритма

6.15. Пример решения задачи средствами ИП

Берем всё тот же пример с обезьяной и бананами. Считаем что модель предметной области уже построена. Следует отобразить ее в терминах ИП. Алфавит *А* ИП, как известно, кроме символов ¬, ∧, ∨, →, ~, ∀, ∃, содержит также:

индивидные константы,
предметные переменные,
функциональные константы,
высказывания,
предикатные константы.

Очевидно, что роль индивидных констант здесь будут игра элементы множества *X* Мпо, т.е. множество имен объектов. У на это Обезьяна (*О*), Ящик (*Я*), Бананы (*Б*). Роль предметных переменных – элементы множества *С* (множество имен свойств объектов). В нашем примере - это координаты Обезьяны, Ящика и Бананов – соответственно *х, у, с*. Область определения для них – множество *D* точек комнаты. В качестве функциональных переменных, как правило, выступают операторы (действия). В нашем случае – множество $G = \{g_1$ – подойти, g_2 – перенести, g_3 – взобраться, g_4 – схватить$\}$. Областью определения функций является множество состояний предметной области, которое возникает в результате выполнения этих действий. Это состояния, являющиеся предусловиями и постусловиями применения действий. Обозначим это множество состояний через $S = \{s_1, s_2, ...s_n\}$.

Множество *R* – множество имен отношений. Например:

НА(О,Я) – Обезьяна *НА* Ящике;
У(О,Я) – Обезьяна *У* Ящика;
$\bar{В}$*(О,Б)* – Бананы *НЕ-В* руках Обезьяны;
и т.п. – готовая предикатная форма записи.

Координаты объектов введем через одноместные предикаты.

О(х) – Обезьяна находится в т. *х*;
Я(у) – Ящик находится в т. *у;*
Б(с) – Бананы висят в т. *с*. Здесь *(х,у,с* ∈*D)*.

Трехместный предикат делает семантику более гибкой:

В(О, Я, х) – Обезьяна и Ящик находятся *В* т. *х*.
У(О,Б,с) – Бананы *У* Обезьяны в т. *с*.

Из сказанного видно, что роль термов у нас играют индивидные константы (*О,Я,Б*) и предметные переменные (*х,у,с*). Но этого недостаточно, т.к. они описывают лишь отдельные состояния предметной области. Переход из одного состояние в другое осуществляется под действием операторов g_i, прилагаемых в данной точке *х* к конкретному состоянию *s*: $g_i(x,s)$. Это тоже терм. Следовательно, допустимы такие выражения (для *х,у,с* ∈*D; s* ∈*S*):

В(О, Я, у, $g_1(x,s)$*)* – "Обезьяна и Ящик находятся в т. *у* в результате применения оператора ***подойти к Ящику*** к состоянию *s* в т. *х*";

НА(О, Я, у, $g_2(y,s)$*)* – "Обезьяна сидит на Ящике в т. *у* в результате применения оператора ***взобраться на Ящик*** к состоянию *s*".

Используя таким образом язык исчисления предикатов, запишем нашу модель предметной области в виде следующих аксиом

1. $\forall x\,\forall s[(\overline{\text{РЯДОМ}}\,(О,Я,x,s))\to В(О,Я,y,g_1(x,s))]$ –

"для всех x и s: если в состоянии s Обезьяна и Ящик **не** находятся рядом в т. x, то Обезьяна может оказаться в т. y, где находится Ящик, путем применения к ситуации s оператора g_1 (***подойти к Ящику***) из точки x в точку y".

2. $\forall y\,\forall s[(В(О,Я,c,g_2(y,s))\to НА(О,Я,c,g_3(c,s))]$ –

"для всех y и s: если Обезьяна и Ящик под действием оператора g_2 (***перенести Ящик***) оказались в точке c, то Обезьяна обязательно заберется на Ящик под действием g_3 (*взобраться на Ящик*)".

3. $(\,\overline{У}(О,Б,c,\text{Sн}))\to НА(О,Я,c,\text{Sн})$ –

"если в точке c у Обезьяны нет Бананов, то она НЕ-НА Ящике".

4. $\overline{\text{РЯДОМ}}\;(О,Я,b,\text{Sн})$ –

" в начальном состоянии Обезьяна и Ящик не находятся рядом".

5. $\forall y\,\forall s(В(О,Я,c,g_2(y,s))$ –

"для всех y и s: Обезьяна и Ящик находятся в точке c в результате применения оператора g_2 (***перенести Ящик***)" к состоянию s.

(В выражениях 3–4 кванторы $\forall$ отсутствуют, т.к. это конкретные высказывания, характеризующие начальное состояние Sн).

Поставим теперь вопрос: существует ли такое состояние $s\in S$ в некоторой точке $x\in D$, при котором Бананы находятся у Обезьяны? Формально вопрос запишется так:

6. $\exists s(У(О,Б,x,s))$,

т.е. мы как бы говорим: "Да, существует такое состояние s". Требуется, таким образом, доказать, что это утверждение истинно. (С точки зрения формальной логики это случится, если выражение 6 окажется логическим следствием пяти предыдущих посылок (1.2.3.4.5)$\mapsto$ 6.

Как и прежде, будем использовать метод опровержения и, как следствие, *метод резолюций*. Для начала приведем выражения, имеющие кванторы, к *предваренной* нормальной форме, затем возьмем отрицание от формулы 6. В последнем случае получим цепочку преобразований:

$$\overline{\exists}s(У(О,Б,x,s)) = \forall s(\,\overline{У}(О,Б,x,s) = \,\overline{У}\,(О,Б,x,s).$$

Добавим полученное выражение к пяти вышеприведенным формулам, представленным в предваренной форме. Получим следующую систему предложений:

1'. $\text{РЯДОМ}(О,Я,x,s)\vee В(О,Я,y,g_1(x,s))$;

2'. $\overline{В}(О,Я,c,g_2(y,s))\;\vee НА(О,Я,c,g_3(c,s))$;

3'. $У(О,Б,c,\text{Sн})\;\vee\overline{НА}\,(О,Я,c,\text{Sн})$;

4'. $\overline{\text{РЯДОМ}}\,(О,Я,b,\text{Sн})$;

5'. $В(О,Я,c,g_2(y,s))$;

6'. $\overline{У}(О,Б,x,s)$.

Система из пяти предложений 1'–5' непротиворечива по построению (можно проверить). Если теперь путем подстановок и резолюций мы определим пустой дизъюнкт в расширенной системе 1' – 6', то это будет означать, что добавление 6' приводит ее к противоречию. Но 6' есть отрицание выводимого нами утверждения 6, и поэтому само выражение 6 приводить к противоречию уже не будет (что-нибудь одно!). И, следовательно, выводимо.

Делаем подстановки, строим резольвенты.

Резольвенты		Подстановки
7'. $НА(О,Я,c,g_3(c,s))$,	(2', 5')	-
8'. $\overline{НА}\,(О,Я,c,\text{Sн})$,	(3', 6')	c/x; $\text{Sн}/s$.
9'. "Л".	(7', 8')	$\text{Sн}/g_3(c,s)$.

Нашли "пустой" дизъюнкт. Это значит, что наше утверждение $\exists\, s(У(О,Б,x,s))$ верно: "Да, существует такое состояние s, при котором Бананы в руках у Обезьяны".

Возможны другие пути. Например:

7". (1', 4'), 9". (3', 8"),

8". (7", 2'), 10". (6', 9"). И т.п.

В качестве упражнения предлагаем раскрыть эти резольвенты, разумеется, с учетом подстановок.

Упражнения

Определите, какие из нижеприведенных формул общезначимы, а какие невыполнимы.

а. $\overline{\forall}xP(x)\wedge\exists yP(y)$;

б. $\exists xP(x)\to\overline{\forall}yP(y)$;

в. $\overline{\forall}xP(x)\wedge\overline{\exists}yP(y)$;

г. $\forall x P(x) \wedge \exists x \; \overline{P}(x);$

д. $\exists x P(x) \to \exists y P(y);$

е. $\forall x P(x) \vee \exists y \; \overline{P}(y).$

2. Привести к предваренной форме, сделать преобразования и доказать следующие тождества.

а. $\forall x P(x) \to (\exists y P(y) \to (\forall x P(x) \wedge \forall z P(z)));$

б. $(\exists x P(x) \wedge \forall x P(x)) \to \forall y P(y);$

в. $\exists x Q(x) \to (\forall y P(y) \to \forall z Q(z));$

г. $\exists x Q(x) \to (\forall x Q(x) \vee \forall y P(y));$

д. $(\exists x P(x) \to \forall y Q(y)) \to ((\forall x P(x) \to \exists z\text{-}Q(z)) \to \forall u \; \overline{P}(u));$

е. $(\overline{\forall} x P(x) \vee \overline{\exists} y Q(y)) \to (\exists x P(x) \to \overline{\exists} z Q(z));$

ж. $\forall x (P(x) \to Q(x)) \to (\exists x P(x) \to \exists x Q(x));$

з. $(\forall x P(x) \vee \forall y Q(y)) \to \forall z (P(z) \vee Q(z)).$

3.Доказать тождества методом опровержения.

а. $\exists x (P(x) \wedge Q(x)) \to (\exists y P(y) \wedge \exists z Q(z));$

б. $\forall x (P(x) \to Q(x)) \to (\exists x P(x) \to \exists z Q(z));$

в. $\exists y \forall x P(x,y) \to \forall x \exists y P(x,y);$

г. $\overline{\exists} x P(x) \to \overline{\forall} y P(y);$

д. $\forall x P(x) \to ((\exists y P(y) \to \forall y Q(y)) \to \exists z Q(z)).$

4..Докажите нижеследующие силлогизмы.

А.. Кто мяукает, тот кошка.
Собаки – не кошки.
Следовательно: собаки не мяукают.

Б. Без свободы нет счастья.
Без счастья нет любви.
След.: «Где нет свободы, быть не может любви».

В. Моряки – сильные люди.
Сильные люди не плачут.
Мишка – моряк,
След.: он не плачет.

ЗАКЛЮЧЕНИЕ

Изложение материала в данном пособии следовало естественной логике изучаемой науки. Поэтому начиналось всё с изучения основополагающих понятий, таких как предметная область, модель предметной области, методы поиска в пространстве состояний, решение задач путем разбиения на подзадачи и т.п.

Далее следовал вопрос о знаниях и методах их представления, т.е. как раз то, что составляет основу систем искусственного интеллекта. Но "поднять" такой материал в рамках одного пособия не представлялось возможным, в то время как непосредственный переход в формальным системам оказывался малообъяснимым. Это и послужило причиной появления главы о методах представления знаний, содержащей краткий и не совсем полный обзор этих методов в самом простом их изложении.

Последующие главы посвящены изучению математических основ ИИ - исчислению высказываний и исчислению предикатов. Материал этих глав довольно сложен, имеющаяся литература мало соответствует институтскому курсу ИИ. Поэтому стояла задача по возможности упростить изложение, иллюстрируя его многочисленными примерами не в ущерб полноте и стройности. Таково содержание первой части курса по СИИ.

Авторы планируют продолжить работу над пособиями по данной теме. Во второй части будут рассмотрены семантические модели представления знаний, в частности, продукции, семантические сети и фреймы, а также проблемы логического вывода на этих моделях. Существенное внимание будет уделено псевдофизическим логикам - времени, размытым множествам, правдоподобному выводу.

Третья часть будет полностью посвящена вопросам разработки интеллектуальных систем, в особенности - экспертных (ЭС). Это вопросы технологии проектирования ЭС, интеллектуальный интерфейс, методы объяснения решений и пополнение знаний. Будут рассмотрены примеры ЭС, эволюция их развития и современное состояние, а также вопросы программно-аппаратной поддержки СИИ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Нильсон Н. Принципы искусственного интеллекта Пер.с англ. - М.: Радио и связь. 1985. - 376 с.

2. Попов Э.В. Экспертные системы. - М.: Наука, 1987.

3. Построение экспертных систем / Под ред. Ф.Хейес-Рот, Д.Уотерман, Д.Ленат - М.: Мир, 1987.

4. Нильсон Н. Искусственный интеллект. - М.: Мир,1973

5. Лорьер Ж.Л. Системы искусственного интеллекта – М.: Мир. 1991.

6. Кондрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А. Представление знаний о времени и пространстве в интеллектуальных системах / Под ред. Д.А.Поспелова. - М.: Наука. Гл.ред.физ.-мат. лит. 1989 г. - 328 с.

7. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. - М.: Наука, 1983.

8. Уотерман Д. Руководство по экспертным системам - М.: Мир, 1989.

9. Поспелов Д.А. Логико-лингвистические модели в системах управления. - М.: Энергия, 1981.

10. Минский М. Фреймы для представления знаний. - М.: Мир, 1979.

11. Поспелов Д.А. Ситуационное управление: теория и практика. - М.: Наука, 1986.

12. Практическое введение в технологию искусственного интеллекта и экспертных систем / Р.Левин, Д.Дранг, Б Эделсон / Пер. с англ. - М.: Финансы и статистика. 1991.- 239 с.

13. Логический подход к искусственному интеллекту. / Тейз А., Грибомон П. и др. - М.: Мир.1990.

14. Нейлор К. Как построить экспертную систему / Пер. с англ. – М.: Энергоатомиздат, 1991. - 286 с.

15. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. - М.: Радио и связь, 1989. - 1989. – 184 с.

16. Ньюэлл А, Саймон Г. GPS - программа, моделирующая процесс человеческого мышдения. - В кн. "Вычислительные машины и мышление" - М.: Мир. 1967. - с. 283-301.

17. Ньюэлл А., Шоу Дж., Саймон Г. Эмпирические исследования машины "Логик-теоретик": Пример изучения эвристики/ В кн. "Вычислительные машины и мышление".- М.: Мир, 1967 г.

18. Поспелов Д.А. Ситуационное управление. Теория и практика. - М.: Наука, 1986 г.- 284 с.

19. Поспелов Д.А. Логико-лингвистические модели в системах управления, - М.: Энергоиздат, 1981 г.- 230 с.

20. Клыков Ю.И. Ситуационное управление большими системами. - М.: Энергия, 1974 г. - 134 с.

21. Загадская Л.С. Теория систем и методы системного анализа в управлении и связи. - М.: Радио и связь, 1983 г.-с. 109-114, 230-234.

22. Гелернтер Г. Реализация машины, доказывающей геометрические теоремы. В кн. "Вычислительные машины и мышление". - М.: Мир, 1967 г. с. 145-165.

23. Депман И.Я. Первое знакомство с математической логикой. – Л.: Знание, 1963 г. – 56 с.

24. Формальная логика / Под ред. И.Я.Чупахина, И.Н.Бродского – Л.: Изд. ЛГУ , 1977 г. – 358 с.

25. Приобретение знаний. Пер. с япон. / Под ред. С.Осуги, Ю.Саэки. – М.: Мир, 1990. – 304 с.

26. Представление и использование знаний. Пер. с япон./ Под ред. Х. Уэно, М. Исудзука. – М.: Мир, 1989. – 220 с.

27. Амамия М. Танака Ю. Архитектура ЭВМ и искусственный интеллект. Пер. с япон. – М.: Мир, 1993. – 400 с.

28. Осуга С. Обработка знаний. Пер. с япон. 1989.- 293 с.

29. Микони С.В. Модели и базы знаний: Учебное пособие. – СПб: Петербурский гос. Университет путей сообщения, 2000. – 155 с.

СОДЕРЖАНИЕ