Р. А. Ашинянц

Логические методы в искусственном интеллекте

$A \to B$

$\tilde{A} = \{ < \mu_A(x) | x >, x \in X \}$

$A \wedge B$

$\Box(A \to \Diamond(B \to C)) \to \Diamond(B \to (\Box A \to \Diamond C))$

$\neg A_1 \vee \ldots \vee \neg A_n \vee B$

$\mu_{A \vee B}(x) = \max(\mu_A(x), \mu_B(x))$

$\forall x: \dfrac{A(x), A(x) \to B(x)}{B(x)}$

$\neg C \wedge \neg D$

Москва 2001

Р.А. Ашинянц

ЛОГИЧЕСКИЕ

МЕТОДЫ

В ИСКУССТВЕННОМ

ИНТЕЛЛЕКТЕ

Издание второе исправленное, дополненное

Рекомендовано Министерством образования
Российской Федерации
в качестве учебного пособия
для студентов высших учебных заведений,
обучающихся по специальностям группы
"Информатика и вычислительная техника "

Москва 2001

УДК 681.3.016

Рецензенты:
Научно-методический Совет по группе специальностей 220200
Министерства образования РФ,
Кафедра кибернетики Московского института электроники и математики (технический университет),
к.т.н., доц. Волченков Н.Г. - кафедра кибернетики Московского Государственного инженерно-физического институт (технический университет).

Ашинянц Р.А.
Логические методы в искусственном интеллекте
Учебное пособие для вузов. -М.:МГАПИ,2001.-223с, ил.
Издание 2-е исправленное, дополненное

ISBN 5-88538-071-6
ISBN 5-8068-0167-5

В книге систематизировано приведены основные логические модели, используемые в прикладных информационных системах и, в том числе, в системах искусственною интеллекта: логики высказываний и предикатов первою порядка, модальная, немонотонная логики и элементы нечеткой. Через все главы проведены фундаментальные понятия интерпретации, модели, резолюции, необходимые для понимания механизмов логического вывода и рассуждений. К материалу каждой главы предложены задачи, к некоторым из них даются решения.

Учебное пособие предназначено для студентов технических вызов, обучающихся по специальностям группы 220200

"Информатика и вычислительная техника", может оказаться полезной студентам, обучающимся по специальности "Прикладная математика" и аспирантам.

ОГЛАВЛЕНИЕ

При написании книги, которая охватывала бы проблемы искусственного интеллекта (ИИ), возникает целый ряд трудностей. Несмотря на относительную новизну этих проблем, число работ, так или иначе связанных с ними, довольно велико, и составление даже краткого обзора этих работ заняло бы много места и потребовало бы большого времени. Следует учесть и еще одно важное обстоятельство. Для решения одних и тех же задач создаются различные методы, которые с течением времени либо забываются, либо сосуществуют. Кроме того, различные методы, использованные в решении одних и тех же задач, привлекают теории из смежных областей знаний. Если учесть еще, что самих задач также немало, то можно представить состояние специалиста, пытающегося разобраться в проблемах ИИ.

Вряд ли существует общепринятое "канонизированное" определение ИИ. Но бесспорным является ясность в понимании предмета исследования - любая интеллектуальная деятельность человека, подчиняющаяся заранее неизвестным законам. Если предметом информатики является обработка информации, то к области ИИ относятся такие случаи обработки, которые *не могут* быть выполнены с помощью простых и точных алгоритмических методов. Лаконичное определение, сделанное французским исследователем в области ИИ Ж.-Л. Лорьером, обобщает сказанное: "Всякая задача, для которой *неизвестен* алгоритм решения, априорно относится к ИИ" [23] (Прим.: Определение Лорьера требует уточнения: "отсутствие алгоритма” решения означает лишь необходимость выбора между возможными вариантами в пространстве решений). Отсюда и диапазон практически важных задач, которые входят в компетенцию ИИ: обработка и понимание естественного языка, распознавание речи и ее осмысление, экспертные системы различного назначения, в том числе обучающие системы, в которые включается механизмы понимания предмета, а не запоминания, системы распознавания изображений, доказательство теорем, анализ журнальных статей и т.д. Как видим, к сфере ИИ относятся те весьма различные области, где мы действуем, не имея абсолютно точного метода решения проблем, обладающих двумя особенностями:

- информация используется в символьной форме;
- наличие *выбора решения*.

Для решения перечисленных выше задач разнообразные методы. Вместо рассмотрения одних и тех же задач различными методами в настоящей публикации делается попытка рассмотреть различные задачи с позиций одного, *логического* подхода. Логика обладает точной, простой и декларативной семантикой, поскольку она не является системой программирования. Более того, как показывает пример силлогизмов, логические рассуждения относительно близки к обычным рассуждениям человека. С другой стороны, процессор вывода на базе других формализмов может извлекать заключения, которые иногда не могут интерпретироваться логикой. Такие случаи характерны для продукционных систем, особенно при наличии в правилах отрицания, когда процессор вывода может порождать необоснованные с точки зрения логики результаты [39].

Центральная проблема ИИ - проблема представления знаний - может пониматься как последовательность задач переводов: перевод с естественного языка (в том числе и разговорной речи) и формальных языков приводит к представлению знаний в языке формальной логики с последующим переводом на понятный для ЭВМ формализм, т.е. язык программирования. Предполагается, что в ИИ представление знаний должно быть активным процессом, который не только должен обладать способностью к запоминанию, но и должен располагать механизмом, позволяющим извлекать запомненные знания, и проводить рассуждения на их основе. Язык программирования, расположенный на последнем месте в предложенной схеме последовательных переводов, является языком, на котором представляются прикладные программы.

Определив направляющей линией логику в изучении концепций и методов ИИ при использовании их в приложениях, мы базируемся на языке программирования Пролог, наиболее адекватно отражающем формальную логику и рассуждения, выступающим интерфейсом между промежуточной логической формой представления смысла естественного языка и приложениями. Такой подход оказывается плодотворным в описании и использовании реляционных баз данных и определении интерфейса между естественным языком и базой данных. Декларативный формализм Пролога в представлении знаний выгоден для точной характеризации базы знаний. В самом деле, содержимое базы знаний определяется логической теорией, аксиомы которой получаются с помощью синтаксических преобразований правил базы знаний, в то же время ответы, выводимые процессором логического вывода, суть последовательные следствия.

Однако большинство актуальных практических задач, например, создание экспертных систем, вряд ли может быть успешно решено с помощью существующих версий языка Пролог. При всех его достоинствах необходимо включить в него элементы объектного и функционального языков (Прим.: Не задолго до выхода в свет настоящею второго издания обнародована объектно-ориентированная версия Пролога), в частности, возможность наследования свойств объектов и их атрибутов, представления понятия множества, важного в нечетких системах. Кроме того, известная своей неэффективностью процедура возвратов, может быть также улучшена за счет введения так называемой системы поддержки рассуждений, запоминающей промежуточные выводы для последующего их использования при аналогичных вопросах к системе.

Настоящая публикация автором задумана как учебное пособие для студентов технических вузов специальностей группы 22, по курсу "Системы искусственного интеллекта". При многообразии публикаций на эту тему учебное пособие требует отбора материала и выбора способов его изложения. В то же время многолетний опыт преподавания показал, что без минимальной логической культуры проблемы ИИ для студентов остаются абстракциями. В этой связи книга нацелена на минимальную теоретическую подготовку студентов к дальнейшему изучению и проектированию баз знаний, экспертных систем, обучающих систем и т.д. В книге изложены основы логики высказываний (гл. I), логики предикатов первого порядка (гл. II). Особое внимание уделено методу резолюции, на базе которого построен вывод в Прологе.

Глава III знакомит с модальной логикой. Автор по мере возможности старался выдержать стилистику изложения, используя уже привычные представления и понятия, полученные из первых двух глав. В частности, выводимость в модальной логике здесь также базируется на методе доказательства от противного, несколько видоизмененном только в части изображения деревьев вывода.

В четвертая глава посвящена модифицируемым рассуждениям. Материал этой главы, на мой взгляд, способствует пониманию проблем интеллектуализации баз данных, экспертных систем, нетрадиционного взгляда на управляемый возврат в логическом выводе.

В пятой главе вводятся понятия нечетких систем, представление знаний в условиях расплывчатых рассуждений. На наш взгляд, представленный здесь модифицированный принцип резолюции для нечетких систем в отечественной литературе не освещался, хотя метод дает "четкую" трактовку нечеткого вывода. Надеемся, что этот пробел этой публикацией устранен.

Таким образом, книга последовательно ведет от четкой логики к логикам возможного и необходимого и к нечеткой логике. Этот путь, представляется, необходимо пройти для понимания идей ИИ и решаемых задач. Автор пытался изложить все вопросы "настолько просто, насколько возможно, но не проще".

Автор не ставил целью рассмотреть даже какую-то представительную часть задач ИИ. Вряд ли это уместно и возможно в книге, которая имеет своего адресата и несколько иные цели. С другой стороны, чрезвычайно интересные вопросы, входящие в круг проблем ИИ, ждут методического изложения для этого же
адресата.

В настоящем втором издании расширена первая и пятая главы, добавлена новая четвертая глава -"Модифицируемые рассуждения". Автор благодарит доцентов Московского института электроники и математики (технический университет) Д.П. Боголюбова и Н.А. Шимко за критические замечания и поддержку при написании книги, а также всем любознательным студентам МГИЭМ, "выловившим" досадные опечатки.

Москва, 2000г.

Ашинянц Р.А.

ГЛАВА 1: Исчисление высказываний.

1.1. Предложения и высказывания. Формальный характер рассуждении.

В естественном языке выделяют повествовательные, повелительные и вопросительные предложения. Выделим такой тип повествовательных предложений, в которых достаточно определенно формулируется некоторая информация о объектах реального мира, явлениях, событиях и т.д. Такие предложения будем называть высказываниями. **Высказывание** - повествовательное предложение, в котором содержится информация об объекте и которое может быть оценено как *истинное* или *ложное*.

Например, предложение "Николай изучает ПРОЛОГ и любит путешествовать" является высказыванием, а предложение "Берегите лес от огня!" - нет. Первое истинно, если Николай (конкретный) действительно изучает ПРОЛОГ, одновременно являясь любителем путешествий, и ложно в противном случае, Об истинности или ложности второго предложения говорить бессмысленно.

Другой тип предложений - предложения, в которых формулируются задачи: "Докажите, что если p имплицирует q, то ¬q имплицирует ¬p". Предложение повелительного типа, в котором предписывается получить результат. Наконец предложение типа "Умножить число 5 на 12 и результат разделить на 3" по существу является программой действий. Таким образом следует отличать высказывания от задач и программ. Итак, отличительной чертой высказываний состоит в том, что они могут быть истинными или ложными. В логической семантике понятие истинности находит свое уточнение и пользуется для обоснования правильности рассуждений.

Что же такое правильное рассуждение? Приведем пример. Существует предание, что Александрийскую библиотеку сжег халиф Омар. Свое деяние он обосновал с помощью следующего рассуждения: "Если ваши книги согласны с Кораном, то они излишни. Если они не согласны с Кораном, то они вредны. Но вредные и излишние книги следует уничтожить. Значит ваши книги следует уничтожить".

В примере первые три предложения являются посылками, а четвертое - заключением. Рассуждение есть переход от посылок к заключению. Правильно ли рассуждения в приведенном примере, независимо от достоверности предания? С логической точки зрения рассуждения совершенно правильны. Иное дело, что сами посылки не истинны.

Другой пример: "Дикари раскрашивают свое лицо и тело. Некоторые современные женщины раскрашивают свое лицо. Следовательно, некоторые современные женщины - дикари". Очевидно, что данное рассуждение неверно, несмотря на то, что используемые посылки и сделанное заключение можно признать истинными.

Одной из основных задач логики - выявить, какие способы рассуждений правильные, а какие - нет. Существуют рассуждения достоверные (их называют еще *дедуктивными*) и рассуждения правдоподобные (*вероятностные*). Заключение, полученное с помощью дедуктивного рассуждения, достоверно. Результаты вероятностных рассуждений являются гипотетическими. Критерием правильности дедуктивного рассуждения является принцип: правильное рассуждение гарантирует истинность заключения при истинности посылок.

Рассуждая на естественном языке в математике, мы применяем некоторые стандартные формы записи и преобразования информации. Впервые это обстоятельство нашло выражение в логике Аристотеля. Он рассматривал в качестве стандартных форм четыре вида категорических утверждений:

"Все S суть P";
"Некоторые S суть P";
"Все S не суть P";
"Некоторые S не суть P".

Аристотелем была разработана теория рассуждений (умозаключений), в которой посылки и заключения формулируются в виде перечисленных категорических утверждений.

По мере развития логики были уточнены и использованы другие формы высказываний, базирующихся на структуре сложных высказываний. Под сложными высказываниями понимаются такие высказывания, которые образованы из простых, связанных с помощью логических связок "и", "или", "если... то..." и др.

В дальнейшем логика стала исследовать способы рассуждений, основанные на более сложной структуре высказываний. Например, высказывание "Иван старше Петра" не укладывается в формализм Аристотеля. Здесь выражена информация о том, что между Иваном и Петром существуют некоторые отношения. Следовательно необходимо выделить более общий тип высказываний, в которых устанавливаются отношения между объектами, чем высказывания о присущности или неприсущности свойств объектам. Одновременно в логику были введены кванторы общности и существования, применимые не только в категорических высказываниях, но и в высказываниях об отношениях.

Кардинальный сдвиг в анализе стандартных форм рассуждений произошел тогда, когда для построения логической теории был применен метод построения формальных систем с помощью специальных символических языков.

1.2. Понятие формальной системы.

Всякая система состоит из некоторого множества первичных (базовых) элементов, обладающих определенными свойствами. Имея исходные описания, можно логическим путем вывести описание новых свойств, при этом утверждение о наличии исходных или выведенных свойств воспринимают как истинные на основании смысла определений данных элементов.

Для того, чтобы сделать высказывание о системе(мире), необходимо описать объекты этой системы. В формальной системе (ФС) символы, обозначающие объекты, представляют элементы, которые согласно определенным правилам образуют выражения. Истинность полученных выражений устанавливается в связи с возможными интерпретациями входящих в них знаков (приложениями).

Мы будем иметь дело с системами, которые содержат определенное число заранее выбранных и фиксированных *общезначимых* высказываний, называемых **аксиомами**, а ФС -**аксиоматическими**.

ФС называется **простой**, если для любой реальной системы, являющейся интерпретацией данной аксиоматической системы, можно использовать одно и то же число исходных допущений, необходимые для получения тех или иных выводов любой системы.

ФС называется **эффективной**, если каждый вывод аксиоматической системы может быть автоматически перенесен на любую из ее *интерпретаций*.

ФС считается заданной, если выполнены следующие условия.

1. Задано некоторое множество конечного или бесконечного числа элементов, которые называются **термами**. Имеется другое конечное множество, элементы которого являются *связками* или *операциями*.
2. Любую линейно упорядоченную совокупность термов и операций назовем **формулой**. Из множества формул выделим подмножество *правильно построенных формул* (ППФ). Для ППФ задаются их правила конструирования, т.е. определяется эффективная процедура, с помощью которой по данному выражению выясняется, является ли формула правильно построенной (т.е. ППФ) или нет в данной ФС.
3. Выделяется некоторое подмножество ППФ, называемые **аксиомами** ФС. Так же, как и для ППФ для аксиом должна иметься процедура, позволяющая определить, является ППФ аксиомой или нет.
4. Имеется конечное множество $R_1, R_2, ..., R_K$ отношений между ППФ, называемых **правилами вывода**. Должна иметься эффективная процедура, позволяющая для произвольной конечной последовательности ППФ решить, может ли каждый член этой последовательности быть выведен из одной или нескольких предшествующих ППФ этой последовательности с помощью некоторого фиксированного числа правил вывода.

Выводом в ФС называется любая последовательность $A_1, A_2, \ldots, A_n$ такая, что для любого i $(1 \le i \le n)$ ППФ A_i есть либо аксиома ФС, либо непосредственное следствие каких-либо предыдущих ППФ по одному из правил вывода.

ППФ B называется **теоремой** ФС (или выводимой), если существует вывод в ФС, в котором последней ППФ является B. Доказательством (или выводом) теоремы называется последовательность из аксиом, правил вывода и уже доказанных теорем, позволяющих получить данную теорему. Понятие теоремы не обязательно эффективно, так как может не существовать эффективной процедуры, позволяющей определять по данной ППФ, существует ли ее вывод в данной ФС. ФС, для которой такая эффективная процедура существует называется **разрешимой**, в противном случае - **неразрешимой**.

ППФ B выводима из ППФ $A_1, A_2, \ldots, A_n$ (или является следствием $A_1, A_2, \ldots, A_n$) тогда и только тогда, когда существует такая конечная последовательность ППФ $B_1, B_2, \ldots, B_r$, что B_r есть B и для любого i $(1 \le i \le r)$ B_i есть:

1. либо аксиома,
2. либо A_i $(1 \le i \le n)$,
3. либо непосредственное следствие некоторых предыдущих ППФ по одному из правил вывода.

Элементы последовательности ППФ $A_1, A_2, \ldots, A_n$ называются **посылками** вывода (или **гипотезами**). Сокращенно вывод B из $A_1, A_2, \ldots, A_n$ запишем в виде $A_1, A_2, \ldots, A_n \vdash B$, или если $\Gamma=\{A_1, A_2, \ldots, A_n\}$, то $\Gamma \vdash B$ вывод ППФ B без использования посылок есть доказательство ППФ B, а сама B -теорема, и это записывается $\vdash B$.

Приведем несколько свойств понятия выводимости из посылок.

1. Если $\Gamma \supseteq \Pi$ и $\Gamma \vdash B$, то $\Pi \vdash B$. Это значит, что если ППФ B выводима из множества посылок, то она также будет выводима, если к Г добавятся новые посылки.
2. $\Gamma \vdash B$ тогда и только тогда, когда в Г существует конечное подмножество П, для которого $\Pi \vdash B$. Это очевидно из п. 1.
3. Если $\Pi \vdash A$ и $\Gamma \vdash B$ для любой ППФ B из множества П, то $\Gamma \vdash A$. Это значит, что если ППФ A выводима из П и каждая содержащая в П ППФ выводима из Г, то ППФ A выводима из Г.

Таким образом, любая формальная система задается четверкой:

<T,H,A,R>, где

T - множество базовых элементов (алфавит);
H - множество правил конструирования ППФ;
A - система аксиом;
R - множество правил вывода.

Множество T состоит из конечного или счетного множества элементов (термов, операций), из которых по правилам H строится синтаксически правильные совокупности (ППФ) формальной системы. Множеством аксиом может быть объявлено любое множество синтаксически правильных совокупностей. Применение семантических правил R (правил вывода) к элементам множества A позволяет строить семантически правильные совокупности (выводимые формулы). Соотношение между описанными множествами приведено на рис. 1.1. На рис. 1.2 изображена схема построения формальной аксиоматико-дедуктивной системы.

Рис. 1.1.

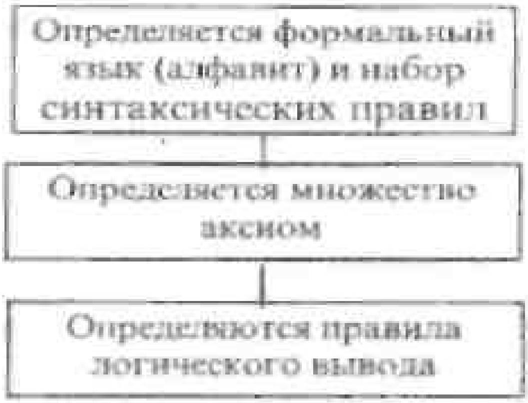


Рис. 1.2.

Всякое исчисление содержит правила вывода, необходимые для его формулировки, они называются **основными**. К основным относятся известные со времен Аристотеля следующие правила и обобщения:

Modus ponens (m.p.) - означает: если из A следует B и A истинно, то и B истинно;

Modus tollens (m.t.) - означает: если A и B не могут одновременно быть истинными и A истинно, то B ложно.

Укрупнение шагов вывода может быть достигнуто за счет применения *производных* правил. Производным правилом вывода называется правило, которое можно построить из посылки с помощью основных правил и, возможно, аксиом данного исчисления.

Правила вывода делятся на прямые и непрямые. **Прямые правила вывода** -это правила непосредственного перехода от одних формул к другим, т.е. от посылки к заключению. Им сопоставляются определенные шаги формального вывода. **Непрямые правила вывода** суть правила перехода от одних формальных выводов к другим. Таким правилам соответствуют метаутверждения о преобразованиях одних формальных выводов в другие.

Еще одним интересным способом рассуждения который может быть оформлен в виде непрямого производного правила, является *метод доказательства от противного*. Пусть нам нужно доказать вывод формулы A из посылок Г. Тогда данную задачу сводят к следующей: отрицание формулы A добавляют к множеству Г и пытаются получить из посылок ¬A, Г противоречие. Если такое противоречие получено, например мы выводим и B, и ¬B, то это означает, что можно построить вывод A из Г. Обозначив противоречие константой можно записать соответствующие этому типу рассуждений правило:

из ¬A, Г ├ f следует Г ├ A.

Заметим, что способ рассуждения от противного, который выражен приведенным правилом является сугубо классическим. Существует аналогичный, но более слабый способ, который можно выразить правилом :

из A, Г ├ f следует Г ├ ¬A

1.3. Логика высказываний.

Высказывание - повествовательное предложение (декларативная фраза), о котором можно сказать, истинно онс или ложно. Значения "истинно" и "ложно" обычно обозначают 1 или 0, или **И** и **Л**, или T и F. Высказывания составляют элементарные фразы логического языка, называемые **атомарными** формулами или **атомами**. Будем обозначать атомарные формулы любыми буквами, возможно с индексами.

Из атомарных высказываний с помощью логических операций строятся сложные высказывания (формулы алгебры высказываний). Правила построения формул определяется *синтаксисом* языка исчисления высказываний:

- всякое высказывание есть формула;
- если X и Y - формулы, то ¬X, (X^Y), (XvY), (X→Y), (X↔Y) -формулы.

Таким образом, синтаксис позволяет выделить формулы из произвольных соединений символов. Первое правило, называемое *базисом*, сопоставляет высказывания висячим узлам, а второе (называемое *индукционным шагом*) порождает дерево, корнем которого является связка, называемая главной связкой формулы. Например, для формулы a^(bvc) имеем дерево:

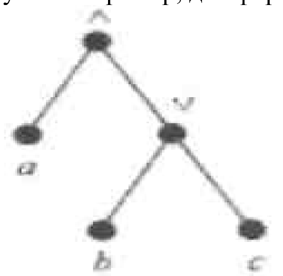


Рис.1.3. Древовидная форма формулы $a \wedge (b \vee c)$

В рассмотренной формуле символы a, b и c являются конкретными высказываниями, прописные же буквы в этой формуле служат для обозначения формулы вообще и называются **мета символами**.

Семантика - это набор правил интерпретации формул. **Интерпретацией** называется приписывание формуле одного из двух значений истинности: **И** (истинно) или **Л** (ложно). *Композиционность* семантики заключается в том, что приписываемое значение истинности некоторой формуле зависит от значения истинности ее составляющих высказываний и структуры формулы.

Интерпретация, при которой формула истинна, называется **моделью** этой формулы. Формула **общезначима**, если она всегда *истинна*, независимо от значений истинности составляющих ее высказываний. **Выполнимой** называется формула, для которой существует хотя бы одна интерпретация, для которой она истинна. Невыполнимую формулу называют **противоречивой**. Общезначимые формулы иначе называют **тавтологиями**. Если A - формула, то запись $\models A$ есть тавтология. Или в более общем виде: если E - множество формул, то $E \models A$ означает, что все интерпретации, обращающие все формулы из E в истинные, обращают в истинную и формулу A. При этом A является логическим следствием E. Таким образом, тавтология является логическим следствием из пустого множества. Примеры, поясняющие данные определения, будут приведены позднее.

Фундаментальная проблема логики, называемая *проблемой дедукции*, состоит в том, чтобы определить, является ли формула A логическим следствием множества E. Само слово **дедукция** (лат. *deductio* - выведение) определяется как логическое умозаключение от общих суждений к частным или другим общим суждениям. Если логическим следствием из множества формул E является формула A, имеющая значение истинности **Л** (ложь или 0), то говорят, что формула A невыполнима. В этом состоит *принцип дедукции*: формула A является логическим следствием множества E тогда и только тогда, когда $E \cup \{\neg A\}$ *невыполнимо*.

Семантика логических связок определяется таблицей истинности. Таблица для связки из n операндов (высказываний) имеет 2^n строк, или интерпретаций. Ниже приводится таблица истинности связок для n=2.

Таблица 1.1

A	B	¬A	AvB	A^B	A→B	A↔B	A⊕B	A|B	A↓B
Л	Л	И	Л	Л	И	И	Л	И	И
Л	И	И	И	Л	И	Л	И	И	Л
И	Л	Л	И	Л	Л	Л	И	И	Л
И	И	Л	И	И	И	И	Л	Л	Л

Содержательно логические связки обычно интерпретируется следующим образом:

отрицание (инверсия) ¬ (¯) - "не";
дизъюнкция v (+) - "или";
конъюнкция ^ (· , &) - "и";
импликация → (⊃) - "если... то";
эквивалентность ↔ (~) - "тогда и только тогда" (или "эквивалентно");
неравнозначность ⊕ - "исключающее или" (сумма по модулю 2);
штрих Шеффера | - "и-не";
стрелка Пирса ↓ - "или-не";

Примем соглашение о приоритете операций, упорядочив по убыванию: ¬, ^, v, →, ↔.

Высказывания A и B называются **равносильными**, если на одинаковых наборах значений переменных (атомарных высказываний), входящих в высказывание, значения этих высказываний будут совпадать. Запись A≡B будет означать, что высказывания A и B равносильны.

Основные равносильности исчисления высказываний:

1. Коммутативный закон

   AvB≡BvA, A^B≡B^A.
2. Ассоциативный закон

   Av (BvC)≡(Av B) v C, A^ (B ^ C)≡(A ^ B)^C.
3. Дистрибутивный закон

   Av(B^C) ≡ (AvB)^(Av C), A^(BvC) ≡ (A^ B)v(A^C).
4. Закон Де Моргана

   ¬(AvB) ≡ ¬A^¬B, ¬(A^B) ≡ ¬Av¬B
5. Закон идемпотентности

   AvA≡A, A^A≡A.
6. Закон поглощения

   Av(A ^ B)≡A, A^(AvB) ≡ A, Av(¬A^ B)≡AvB.
7. Закон склеивания 7'. Закон неполного склеивания

   (A^B)v(A^¬B)≡A (A^B)v(A^¬B)≡Av(A^B) v (A^¬B)
8. Закон исключенного третьего Av¬A=l

   и закон противоречия A^¬A=0
9. Закон двойного отрицания ¬¬A≡A или в общем случае

$$\underbrace{\neg\neg\ldots\neg A}_{n\ раз} \equiv \left\{ \begin{array}{l} A\text{, если } n - \text{чётное} \\ \neg A\text{, если } n - \text{нечётное} \end{array} \right\}$$

10. Av0≡A, A^0≡0
11. Av1 ≡ 1, A^1 ≡ A
12. ¬0≡1, ¬1≡0
13. A→B ≡ ¬AvB
14. A↔B ≡ (A→B)^(B→A) ≡ (A^B)v(¬A^¬B) ≡ (¬AvB)^(Av¬B)
15. A⊕B ≡ (A^¬B)v(¬A^B)
16. A|B ≡ ¬(A^B) ≡ ¬Av¬B
17. A↓B ≡ ¬(AvB) ≡ ¬A^¬B
18. Контрапозиция A→B ≡ ¬B→¬A.

Необходимо сделать некоторые- дополнения о импликации. Эта связка отражает структуру рассуждений. Первый операнд называется **посылкой** (или **антецедентом**), а второй - **заключением** (или **консеквентом**). Из таблицы истинности следует, что при истинности посылки заключение истинно. Однако, и при ложной посылке заключение может оказаться истинным. Эта единственная связка, обладающая свойством:

- если посылка истинна, истинность импликации совпадает с истинностью заключения;
- значение истинности зависит от двух операндов;
- импликация некоммутативна.

Теперь покажем примеры выполнимых, невыполнимых и общезначимых формул:

высказывание а выполнимо, отрицание высказывания также выполнимо; (avb) и (a^b) - выполнимые формулы: они истинны, в частности, при истинности a и b. Формула (a^¬a) невыполнима. Общезначимые формулы: (av¬a), (a→a), ¬¬a.

Покажем, как использовать таблицы истинности для доказательства общезначимости формулы

$$(a \rightarrow (b \rightarrow c)) \leftrightarrow ((a \wedge b) \rightarrow c)$$

Таблица истинности будет содержать восемь строк - интерпретаций (n=3, 2^3=8). Обозначим левую часть этой формулы буквой А, а правую - В, т.е. имеем A↔B. Таблица истинности примет следующий вид:

Таблица 1.2

a	b	c	b→c	a→(b→c)	a^b	(a^b)→c	A↔B
И	И	И	И	И	И	И	И
И	И	Л	Л	Л	И	Л	И
И	Л	И	И	И	Л	И	И
И	Л	Л	И	И	Л	И	И
Л	И	И	И	И	Л	И	И
Л	И	Л	Л	И	Л	И	И
Л	Л	И	И	И	Л	И	И
Л	Л	Л	И	И	Л	И	И

1.4. Формы представления высказываний.

Одно и то же сложное высказывание может быть представлено в различных формах.

Элементарной дизъюнкцией называется выражение $A_1 v ... v A_i ... v A_n$, где каждое A_i (i=1..n) является либо элементарным высказыванием, либо отрицанием элементарного высказывания и $A_i \neq A_j$ при $i \neq j$. Пустой дизъюнкт - единственный невыполнимый дизъюнкт. Его обычно обозначают □ или **Л**.

Элементарной конъюнкцией называется выражение B_1^B_2^...^B_m, где каждое B_i (i=1..m) является либо элементарным высказыванием, либо отрицанием элементарного высказывания.

Исчисление высказываний занимается только теми логическими соотношениями, которые возникают из факта построения одних высказываний из других (называемых элементарными формулами, или **атомами**), которые уже не анализируются.

В *классическом* исчислении высказываний принято, что всякое неанализируемое высказывание (атом) истинно или ложно, но не то и другое вместе.

Примеры:

1. ¬xvyv¬z - элементарная дизъюнкция
2. x^¬y^¬z - элементарная конъюнкция
3. ¬x - можно считать частным случаем как элементарной конъюнкции, так и элементарной дизъюнкции;
4. x^y v z - не является ни элементарной конъюнкцией, ни элементарной конъюнкцией;
5. ¬(xvy) v z - не является элементарной дизъюнкцией;
6. x^¬(y^z) - не является элементарной конъюнкцией.

Дизъюнктивной нормальной формой (ДНФ) данного высказывания называется равносильное ему высказывание вида $K_1 v...vK_i...vK_S$, где K_i (i=1..s) - элементарная конъюнкция.

Конъюнктивной нормальной формой (КНФ) данного высказывания называется равносильное ему высказывание вида D_1^...^D_i...^D_T, где D_i (i=1..t) - элементарная дизъюнкция.

Алгоритм приведения формулы к нормализованной конъюнктивной форме достаточно прост:

- Сначала исключаются из формул в соответствии с приведенными выше равносильностями связки →, ↔.
- Необходимое число раз применяются правила Де Моргана.
- Применяется правило дистрибутивности.

В процессе приведения исходной формулы необходимо упростить полученную на промежуточном этапе формулу. При этом надо помнить, что дизъюнкты, содержащие противоположные литеры (т.е. высказывание и его отрицание), общезначимы и могут быть опущены. Кроме того, можно опустить повторения одного и того же высказывания в пределах одного дизъюнкта.

На простом примере покажем применение этих правил для приведения формулы к КНФ.

Понятие дизъюнкта используется в логическом программировании и, в частности, в языке Пролог.

Совершенной дизъюнктивной нормальной формой (СДНФ) называется такая ДНФ, в которой каждая элементарная конъюнкция (называемая также конституентой единицы) содержит все элементарные высказывания либо их отрицания по одному разу. Конституенты единицы в СДНФ не повторяются.

Совершенной конъюнктивной нормальной формой (СКНФ) называется такая КНФ, в которой элементарная дизъюнкция (называемая также конституентой нуля) содержит все элементарные высказывания либо их отрицания по одному разу. Конституенты нуля в СКНФ не повторяются.

Каждое высказывание, кроме тавтологии и противоречия, имеет единственную СДНФ и единственную СКНФ.

Тавтология не имеет СКНФ. а противоречие — СДНФ.

Примеры:

1. (x^y)v(x^z)vy - ДНФ;
2. (xv¬y)^(¬xv¬z)^x - КНФ
3. (¬x^y^¬z)v(x^¬y^z)v(x^y^z)v(¬x^¬y^¬z) - СДНФ
4. (x v y v z) ^ (¬x v y v ¬z) ^ (¬x v ¬y v z) - СКНФ.
5. ¬xvy - можно рассматривать как ДНФ и как КНФ.
6. (¬x ^ y ^ z) v (x ^ ¬y ^ z) - является СДНФ.
7. (x v ¬y v ¬z) ^ (w v ¬x v ¬z) - не является СКНФ.
8. (x^y^¬z)v(x^y^z)v(x^y^¬z) - не является СДНФ.

При переходе от ДНФ к СДНФ используются следующие равносильности:

A^1 ≡ A; Bv¬B≡1;

A≡A^(Bv¬B)≡(A^B)v(A^¬B)

Пример:

Привести к КНФ формулу (x→(y→z))→((x^q)→z).

Исключим импликации:

¬(x→(y→z)v((x^q)→z) = ¬(¬xv(¬yvz))v(¬(x^q)vz).

Внесем внешние отрицания в скобки, используя закон Де Моргана:

(¬¬x^¬(¬yvz))v(¬(x^q)vz) = (x^(¬¬y^¬z))v(¬xv¬qvz) = (x^(y^¬z))v(¬xv¬qvz).

Теперь применим дистрибутивный закон и одновременно упростим получаемые выражения, помня замечание об общезначимости дизъюнкта, содержащего противоречие:

(xv(¬xv¬qvz))^((y^¬z)v(¬xv¬qvz)) = (xv¬xv¬qvz)^(yv¬xv¬qvz)^(¬zv¬xv¬qvz).

Первая и третья скобки (дизъюнкты) общезначимы, поэтому И^(yv¬xv¬qvz)^И = ¬xvyv¬qvz. Таким образом, полученная конъюнктивная нормальная форма содержит один дизъюнкт. Эта формула ложна при истинности x, ложности y истинности q и ложности z.

Пример:

xv(¬x^y)v(¬x^¬y^z) = x^(yv¬y)^(zv¬z)v(¬x^y)^(zv¬z)v(¬x^¬y^z) = (x^y^z)v(x^y^¬z)v(x^¬y^z)v(x^¬y^¬z)v(¬x^y^z)v(¬x^y^¬z)v(¬x^¬y^z).

При переходе от КНФ к СКНФ используются следующие равносильности:

Av0 ≡ A; B^¬B ≡ 0; A ≡ Av(B^¬B) ≡ (AvB)^(Av¬B).

Пример:

x^(¬xvy)^(¬xv¬yv¬z) = [xv(y^¬y)v(z^¬z)]^[¬xv¬yv(z^¬z)]^(¬xv¬yvz) = (xvxvz)^(xvyv¬z)^(xv¬yvz)^(xv¬yv¬z)^(¬xvyvz)^(¬xvyv¬z)^(¬xv¬yvz).

При переходе от СДНФ к СКНФ вначале необходимо получить отрицание исходного высказывания за счет выписывания через дизъюнкцию недостающих (до полного перечня) конституент единицы, а затем взять отрицание этого высказывания и выполнить преобразования по закону Де Моргана.

Пример:

f = (x^y^z)v(x^y^¬z)v(x^¬y^z)v(x^¬y^¬z)v(¬x^y^z);

¬f = (¬x^¬y^¬z)v(¬x^¬y^z)v(¬x^y^¬z);

¬¬f = ¬[(¬x^¬y^¬z)v(¬x^¬y^z)v(¬x^y^¬z)] = (xvyvz)^(xvyv¬z)^(xv¬yvz).

При переходе от СДНФ к СКНФ вначале необходимо получить отрицание исходного высказывания за счет выписывания через дизъюнкцию недостающих (до полного перечня) конституент единицы, а затем взять отрицание этого высказывания и выполнить преобразования по закону Де Моргана.

Пример:

f = (x^y^z)v(x^¬y^z)v(x^y^¬z)v(x^¬y^¬z)v(¬x^y^z);

¬f = (¬x^¬y^¬z)v(¬x^¬y^z)v(¬x^y^¬z);

¬¬f = ¬[(¬x^¬y^¬z)v(¬x^¬y^z)v(¬x^y^¬z)] = (xvyvz)^(xvyv¬z)^(xv¬yvz).

При переходе от СКНФ к СДНФ вначале необходимо получить отрицание исходного высказывания за счет выписывания через конъюнкцию недостающих (до полного перечня) конституент нуля, а затем взять отрицание этого высказывания и выполнить преобразования по закону Де Моргана.

Пример:

$f = (x \vee y \vee z) \wedge (x \vee y \vee \neg z) \wedge (x \vee \neg y \vee z);$

$\neg f = (\neg x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z);$

$\neg\neg f = f = \neg[(\neg x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z)] =$
$(x \wedge y \wedge y \wedge z) \vee (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z) \vee (x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge y \wedge z).$

1.4.1. Минимизация сложных высказываний с помощью метода Квайна.

Существуют различные методы минимизации сложных высказываний. Рассмотрим один из них, удобный для алгоритмизации, а следовательно, для переноса процесса минимизации на ЭВМ - метод Квайна. Этот метод использует следующие равносильности:

- неполное склеивание (A^B)v(A^¬B) ≡ (A^B)v(A^¬B)vA;
- поглощение AvA^B ≡ A.

Минимизация выполняется в три этапа:

1. Высказывание из произвольной формы переводится в СДНФ.
2. Исходя из СДНФ, получают сокращенную дизъюнктивную форму (СкДНФ).
3. На основе СДНФ и СкДНФ строится импликантная матрица, с помощью которой получается минимальная дизъюнктивная нормальная форма (МДНФ).

Заметим, что данный метод (как и большинство других) находит минимальную среди дизъюнктивных нормальных форму для данного высказывания (т.е. требование нормальности формы является серьезным ограничением).

Первый этап минимизации был рассмотрен ранее, когда обсуждался переход от ДНФ к СДНФ.

На втором этапе минимизации СДНФ выполняют все возможные операции поглощения. Полученная в результате форма называется **сокращенную дизъюнктивную форму (СкДНФ)**.

На заключительном этапе строится импликантная матрица, столбцами которой являются конституенты единицы исходной СДНФ, строками - элементы СкДНФ, называемые **импликантами**. Выбирается набор импликант, содержащий минимальное количество элементарных высказываний и такой, что совокупность выбранных импликант имеет вхождения во все конституенты единицы. Дизъюнкция этих импликант дает МДНФ.

Примеры:

1. Получить МДНФ для высказывания (x↓y)v((xvy) ^ ¬z).

Начнем с получения СДНФ:

(x ↓ y)v((x v y) ^ ¬z) = (¬x^¬y)v(x^¬z)v(y^¬z) =
(¬x^¬y^z)v(¬x^¬y^¬z)v(x^y^¬z)v(x^¬y^¬z)v(¬x^y^¬z)
1 2 3 4 5

Далее получим СкДНФ. При склеивании будем указывать порядковые номера конъюнкций, участвующих в склеивании, и рядом записывать результат. Неполное склеивание будет обеспечено за счет того, что конъюнкции, участвующие в склеивании, не уничтожаются:

1-2: ¬x^¬y 6 3-5: y^¬z 10
2-4: ¬y^¬z 7 7-10: ¬z
2-5: ¬x^¬z 8 8-9: ¬z
3-4: x^¬z 9

Теперь выполняются поглощения. Результат каждого склеивания поглощает конъюнкции, участвующие в склеивании. Выполнив все возможные склеивания, получим СкДНФ ¬x^¬y v ¬z. Получим МДНФ (табл. 1.3):

Таблица 1.3.

	¬x^¬y^z	¬x^¬y^¬z	x^y^¬z	x^¬y^¬z	¬¬x^y^¬z
¬x^¬y	+	+			
¬z		+	+	+	+

Из импликантной матрицы следует, что МДНФ совпадает с СкДНФ.

2. Получить МДНФ для высказывания ((x|y)→(y↓z))v¬(¬x→¬z).

Получим СкДНФ:

((x | y) → (y ↓ z))v¬(¬x → ¬z) = ¬¬(x^y)v(¬y^¬z)v(¬x^z) = (x^y)v(¬y^¬z)v(¬x^z) =
(x^y^z)v(x^y^¬z)v(x^¬y^¬z)v(¬x^¬y^¬z)v(¬x^y^z)v(¬x^¬y^z)
1 2 3 4 5 6

Получим СкДНФ:
1-2: x^y
1-5: y^z
2-3: x^¬z
3-4: ¬y^¬z
4-6: ¬x^¬y
5-6: ¬x^z

СкДНФ имеет вид:
(x^y)v(y^z)v/(x^¬z)v(¬y^¬z)v/(¬x^¬y)v(¬x^z).
В данном случае получается две МДНФ (табл. 1.4):
МДНФ1: (x^y)v(¬y^¬z)v(¬x^z);
МДНФ2: (y^z)v(x^¬z)v(¬x^¬y).

Таблица 1.4.

	x^y^z	x^y^¬z	x^¬y^¬z	¬x^¬y^¬z	¬x^y^z	¬x^¬y^z
x^y	+	+				
y^z	+				+	
x^¬z		+	+			
¬y^¬z			+	+		
¬x^¬y				+		+
¬x^z					+	+

1.4.2. Правила минимизации с помощью карт Карно.

Строение карты. Карта Карно представляет собой поле клеток, число которых равно числу наборов аргументов 2^l . Столбцы и строки нумеруются частичными наборами в порядке кода Грея (рис. 1.4а) Также широко используется второй способ нотации карты (1.4b). Черточками обозначены группы клеток, в которых данный аргумент равен 1. В неохваченных черточкой клетках этот аргумент равен 0. Расположение букв-аргументов, в принципе, может быть произвольным, но для единообразия всегда будем считать, что x_1 - первый (младший) разряд, а x_4 - старший разряд.

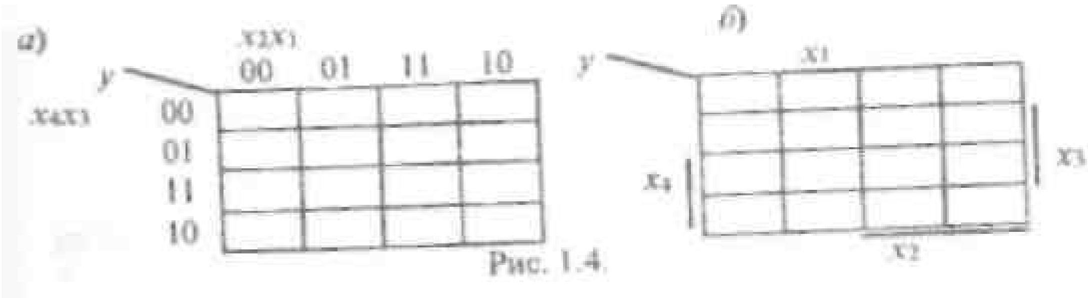


Рис. 1.4.

Каждая клетка находится на пересечении столбца и строки соответствует одному полному набору аргументов (одному из 2^l минтермов). Поэтому клеткам присваиваются номера строк таблицы истинности (рис. 1.5)

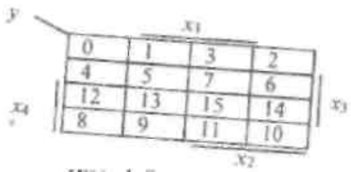


Рис. 1.5.

В клетку записывается 1, если на соответствующем наборе функция y=1. Избыточные клетки заполняются следующими знаками:

x - набор запрещен или никогда не появляется;

Ф - функция безразлична к данному набору;

φ - условно-безразличное значение.

Остальные клетки соответствуют нулевым значениям представляемой функции.

Пример. Недоопределенная функция четырех переменных задана комплексами единичных значений и избыточных наборов

$$K'=\left\{\begin{matrix}0101\\1100\\1001\end{matrix}\right\} \quad x=\left\{\begin{matrix}0000\\0001\\0010\\1101\\1110\end{matrix}\right\}$$

Карта Карно этой функции изображена на рис. 1.6.

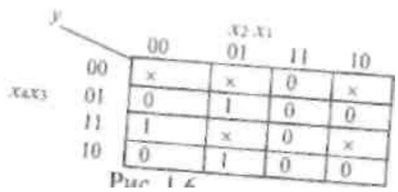


Рис. 1.6.

В простых задачах, когда опасность ошибки исключена, нули можно не записывать, оставляя клетки пустыми.

Общие свойства карты Карно. Соседние клетки отображают соседние минтермы СДНФ. Значит, определенные группы соседних клеток с единицами могут быть подвергнуты операции склеивания сразу. Такие группы обводятся овалами.

Каждая склеиваемая группа клеток с единицами дает одно слагаемое искомой МДНФ, выражаемое только теми аргументами, значениями которых одинаковы для всех клеток данной группы.

Число клеток в группе, склеиваемой сразу, всегда равно целой степени двух.

Общие правила склеивания. Во избежание появления избыточных импликант нужно:

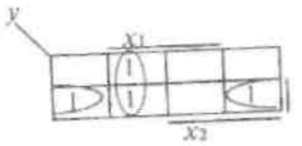
a) прежде всего склеивать группы, содержащие такие клетки с единицами, которые не входят ни в какую другую группу;
b) стремиться к тому, чтобы группа содержала как можно больше клеток с единицами, а число групп было как можно меньшим;
c) обводить овалами склеиваемые группы так, чтобы каждый новый овал охватывал по крайней мере одну новую (еще не охваченную) клетку с единицей.

Примечание. В тех случаях, когда число клеток с единицами превышает половину карты, удобнее склеивать клетки с нулями, определяя первоначально ¬y, а затем инвертировать результат.

Правила образования групп.

А. *Пары клеток с единицами.*

а) Соседними клетками считаются не только две смежные клетки, но и клетки, расположенные на краях одной и той же строки или одного и того же столбца. При наличии в них единиц, они обводятся частями овала (рис. 1.7).



$y = x_1\neg x_2+\neg x_1 x_3$.

Рис. 1.7.

б) Соседние пары огут выбираться в произвольном порядке и каждая клетка может быть использована много раз. Однако неоправданные перекрытия овалов могут привести неминимальной тупиковой ДНФ.

Например, нерациональная группировка (рис 1.8а.) дает $y=x_1\neg x_2+\neg x_1 x_2+x_2\neg x_3+\neg x_2 x_3$, тогда как рациональная (рис 1.8б) приводит к МДНФ $y = x_1\neg x_3+\neg x_1 x_2+\neg x_2 x_3$.

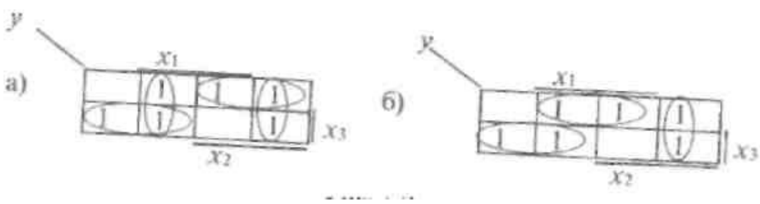


Рис. 1.8.

Б. *Квадраты*. Две соседние пары с единицами составляют группу клеток, называемую "квадратом". Соседними парами считаются не только смежные пары, но и пары, расположенные на противоположных краях карты. Оки обводятся частями овала, охватывающего квадрат. Квадрат также образуется четырьмя угловыми клетками с единицами.

Например, на карте рис. 1.9 имеется три квадрата. Так, что
$y = \neg x_1\neg x_3+\neg x_2\neg x_3+\neg x_2\neg x_4$.

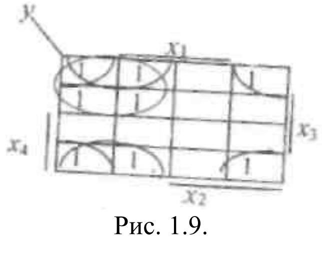


Рис. 1.9.

Нужно избегать избыточных квадратов. Например, каждая клетка квадрата (пунктир на рис. 1.10) уже использована в парах, т.е. этот квадрат избыточен.

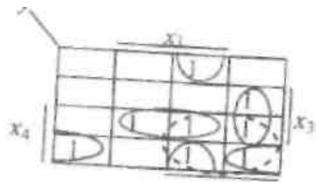


Рис. 1.10.

В. *Целые строки и столбцы.* Если целая строка или целый столбец заполнены единицами, то они склеиваются сразу, давая конъюнкцию только тех аргументов, которые составляют номер склеиваемой строки или столбца. В примере на рис. 1.11 y = x1x2+x3x4.

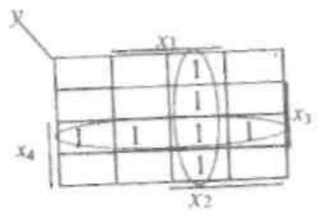


Рис. 1.11.

Г. *Соседние четверки.* Два соседних неперекрывающихся квадрата или две соседних целых строки, а также два соседних целых столбца склеиваются сразу. Соседними также считаются строки и столбцы, расположенные на противоположных краях карты.

Склеивание долей всей карты. Если склеиваются сразу клетки всей карты, то получается <u>константа единица</u>. Если склеивается сразу половина карты, то получается <u>одна переменная.</u> Если склеивается сразу четверть карты, то получаются две переменные и т.д.

1.5. Анализ рассуждений. Прямые методы вывода.

Этот параграф посвящен вопросам использования классического исчисления высказываний в рассуждениях, проводимых средствами естественного языка. Решить логические проблемы, выраженные в словесной форме, можно было бы только, переведя все фразы в символику исчисления высказываний, а затем использовав теорию и аппарат этого исчисления применительно к полученным формулам.

Пример [2]. Я заплатил бы за работу по ремонту телевизора (З), только если бы он стал работать (Р). Он же не работает. Поэтому я платить не буду.

Рассуждение можно записать так:

З→Р, ¬Р ├¬З.

Установим эту формулу, выписывая последовательно формулы, выводимые из З→Р, ¬Р, начиная с них самих, до тех пор, пока не наткнемся на ¬З:

1. З→Р, 2. ¬Р, 3. ¬Р→¬З (из 1 контрапозицией),
4. ¬З (из 2 и 3 по m.p.).

Пример. Если бы он ей не сказал ¬С, она ни за что не узнала бы ¬У. А не спроси она его ¬В, он не сказал бы ей. Но она узнала. Значит, она его спросила.

Символически:

¬С→¬У, ¬В→¬С, У ├В.

Установим, что это рассуждение общезначимо.

В самом деле: 1. ¬С→¬У, 2. ¬В→¬С, 3. У, 4. У→С (контрапозиция из 1), 5. С→В (контрапозиция из 2), 6. С (m.p.. из 3, 4), 7. В (m.p. из 6, 5).

Другой вариант: 4. ¬В¬У (транзитивность импликации и дважды m.p. для 2, 1),
5. У→В (контрапозиция 4), 6. В (m.p. 3, 5).

Заметим, что довод, который зачастую выдается за тривиальное умозаключение в один шаг ("значит"), не получается ни по какому очевидному логическому принципу. Можно было бы на такие рассуждения возразить, что не очевидно, откуда следует заключение, и заключение не следует из посылок. Рассмотрим еще один пример.

Пример. Он сказал, что придет П, если не будет дождя ¬Д. Но идет дождь. Значит он не придет.

Символически: ¬Д→П, Д├¬П.

Для того, чтобы переделать это рассуждение в виде ¬Д→П, Д├¬П, используем m.p. Но для этого посылку надо иметь в виде ¬Д. Если подвергнуть ¬Д→П контрапозиции и упростить затем ¬¬Д до Д, то получим ¬П→Д, что опять ничего не дает.

Пусть Д и П имеют значение "истина". Тогда обе посылки принимают истинное значение, а предлагаемое ¬П принимает значение "ложь". Следовательно, ¬Д→П, Д├П не верно, заключение не самом деле не следует из посылок.

Переводя выражения естественного языка с помощью связок, мы лишаемся некоторых оттенков смысла, но зато выигрываем в точности. Так, известно, что в исчислении высказываний А^В равносильно В^А. Но фразы "Миша открыл дверь и вошел в комнату" и "Миша вошел в комнату и открыл дверь" имеют совершенно разный смысл. В этом примере порядок высказываний в конъюнкции наводит на мысль о следовании во времени или о причинно-следственной связи. Следование во времени можно выразить с помощью символизма исчисления предикатов.

Другой пример возможного искажения смысла при изменении порядка следования высказываний в логических формулах - использование равносильности двух высказываний: AvB и BvA. Но фразы "Я не приду, если она не извинится" и "Она не извинится, если я не приду" конечно же имеют разный смысловой оттенок.

Другая трудность перевода состоит в двусмысленности определенных терминов, когда их надо переводить связками. Например, если в меню кафе указано: "Чай или кофе бесплатно", то ничего удивительного в том, что попросив и чай и кофе, мы получим увеличенный счет.

Двусмысленностей можно избежать, если при переводе фразы в символьную форму воспользоваться исходной формой, а не равносильной ей. Другое дело, что в процессе вывода неизбежно используются всевозможные равносильные преобразования.

Важно в процессе доказательства выводимости формул из аксиом умело использовать правила, которые следуют из классической логики высказываний. В рассмотренных в этом разделе примерах на каждом шаге вывода мы ссылались на определенное правило, следствием чего и стал вывод на данном шаге. Теперь мы можем привести полный перечень правил вывода, который поможет сориентироваться при выборе пути доказательства. Приведенные ниже правила следует трактовать следующим образом: из истинности левой части следует правая часть. Использован знак следования $\vdash$.

В1. Введение конъюнкции: $A, B \vdash A \wedge B$

В2. Удаление конъюнкции: $A \wedge B \vdash B$; $A \wedge B \vdash A$;

В3. Отрицание конъюнкции (закон Де Моргана): $\neg(A \wedge B) \vdash \neg A \vee \neg B$

В4. Введение дизъюнкции: $A \vdash A \vee B$; $B \vdash A \vee B$

В5. Удаление дизъюнкции: $A \vee B, \neg B \vdash A$; $A \vee B, \neg A \vdash B$

В6. Отрицание дизъюнкции (закон Де Моргана): $\neg(A \vee B) \vdash \neg A \wedge \neg B$

В7. Удаление импликации (modus ponens): $A \to B, A \vdash B$

В7'. Удаление импликации (modus tollens): $A \to B, \neg B \vdash \neg A$

В8. Отрицание импликации: $\neg(A \to B) \vdash A \wedge \neg B$

В9. Введение эквивалентности: $A \to B, B \to A \vdash A \leftrightarrow B$;

В10. Удаление эквивалентности: $A \leftrightarrow B \vdash A \to B$;

В11. Введение двойного отрицания: $A \vdash \neg\neg A$

В12. Удаление двойного отрицания: $\neg\neg A \vdash A$

В13. Правило дедукции: $(\Gamma, A \vdash B) \vdash (\Gamma \vdash A \to B)$

В14. Доказательство от противного: $(\Gamma, \neg A \vdash B \wedge \neg B) \vdash (\Gamma \vdash A)$

В15. Сведение к абсурду: $(\Gamma, A \vdash B \wedge \neg B) \vdash (\Gamma \vdash \neg A)$

В правилах В1 - В15 А и В - формулы, а Г- множество формул, возможно пустое.

Пример. Обосновать выводимость $A \to (B \to C), A \wedge B \vdash C$

1. $A \to (B \to C)$ - гипотеза;
2. $A \wedge B$ - гипотеза;
3. A - из 1 по В2;
4. $B \to C$ - из 1 и 3 по В7;
5. B - из 2 по В2;
6. C — из 4 и 5 по В7.

Следовательно, $A \to (B \to C), A \wedge B \vdash C$- по определению вывода на основе 1 — 5.

1.6. Непрямые методы вывода. Метод резолюции.

Метод резолюции - это один из методов доказательства от противного.

Метод, введенный Дж.Робинсоном, является теоретической базой для построения большинства методов доказательства теорем. Робинсон пришел к заключению, что общепринятые правила вывода, например, правило *modus ponens*, позволяют человеку интуитивно проследить за каждым шагом процедуры доказательства. Он открыл более сильное правило *резолюций*, которое трудно поддается восприятию, но эффективно реализуется на компьютере. При использовании правила резолюции можно вывести любое из множеств аксиом, представленных во фразовой форме.

Не существует по-настоящему эффективных критериев проверки *выполнимости* КНФ. Метод резолюции позволяет выявить *невыполнимость множества дизъюнктов*. Действительно, множество дизъюнктов невыполнимо тогда и только тогда, когда пустой дизъюнкт □ является логическим следствием из него. Таким образом невыполнимость множества S можно проверить, порождая логические следствия из S до тех лор, пока не будет получен пустой дизъюнкт.

Для порождения логического следствия используем простую схему рассуждений. Пусть A, B и X - формулы. Предположим, что две формулы C1: (AvX) и C2: (Bv¬X) - истинны. Если X тоже истинна, то можно заключить, что и B истинна. Наоборот, если X ложна, то можно заключить, что A истинна. В обоих случаях (AvB) истинна. Получаем правило:

$\{A \vee X, B \vee \neg X\} \vdash A \vee B$, или в равносильной форме

$\{\neg X \rightarrow A, X \rightarrow B\} \vdash A \vee B$.

Мнемоническое правило вывода можно сформулировать следующим образом: если пара исходных дизъюнктов множества S содержит контрарные литералы (X и ¬X), то новый дизъюнкт формируется из оставшихся частей дизъюнктов, не содержащих эти контрарные литералы. Этот вновь сформированный дизъюнкт называется **резольвентой** исходных дизъюнктов.

Общезначимость правила резолюций выражается следующей леммой.

Лемма 1. Пусть s_1 и s_2 - *дизъюнкты нормальной формы S*, I - *литера*. Если $I \in s_1 и \neg I \in s_2$, то *дизъюнкт* $r=(s_1 \setminus \{l\}) \vee (s_2 \setminus \{\neg l\})$ *является логическим следствием нормальной формы S*.

Если в процессе вывода новых дизъюнктов мы получаем два однолитерных дизъюнкта, образующих контрарную пару, то резольвентой этих двух дизъюнктов будет пустой дизъюнкт □. Таким образом, выводом пустого дизъюнкта □ из множества дизъюнктов S называется конечная последовательность дизъюнктов $C_1, C_2, \ldots, C_K$ такая, что любой C_i является или дизъюнкт из S или резольвентой, полученной принципом резолюции, и Q=□.

Вывод пустого дизъюнкта может быть наглядно представлен с помощью дерева вывода, вершинами которого являются или исходные дизъюнкты или резольвенты, а корнем - пустой дизъюнкт.

Пример. Пусть S = {PvQ, ¬PvQ, Pv¬Q, ¬Pv¬Q}. Пронумеруем исходные дизъюнкты:

1. PvQ.	5. Q.	(1, 2).
2. ¬PvQ.	6. ¬Q.	(3, 4).
3. Pv¬Q.	7. □.	(5, 6).
4. ¬Pv¬Q.		

Вывод представлен на рисунке 1.12с помощью дерева вывода.

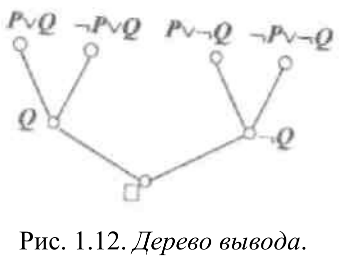


Рис. 1.12. *Дерево вывода*.

Доказательство вывода формулы G из множества посылок $F_1, F_2, \ldots, F_n$ сводится к выводу противоречивости формулы $F_1 \wedge F_2 \wedge \ldots \wedge F_n \wedge \neg G$. Таким образом к исходному множеству посылок (аксиом) необходимо добавить *отрицание* выводимой формулы (теоремы). Все высказывания в задаче необходимо представить в форме дизъюнктов, и используя метод резолюций, вывести пустой дизъюнкт, т.е. доказать невыводимость из множества исходных дизъюнктов отрицание теоремы. Тогда исходное утверждение должно быть верным. В этом и заключается доказательство от противного.

Пример [9]. Если команда А выиграет в футбол, то город A_1 торжествует, а если выиграет команда В, то будет торжествовать город B_1. Выиграет или команда А или команда В. Если выиграет команда А, то город В, не торжествует, а если выиграет В, то не будет торжествовать город А,. Следовательно, город B_1 будет торжествовать тогда и только тогда, когда не будет торжествовать город A_1.

В символическом виде имеем:

1. $A \rightarrow A'$.	4. $A \rightarrow \neg B'$.
2. $B \rightarrow B'$.	5. $B \rightarrow \neg A'$.
3. $A \vee B$.	6. $B' \rightarrow \neg A'$.

Приведем высказывания к форме дизъюнктов. Последнее (шестое) высказывание, истинность которого надо доказать, представим в форме отрицания:

$\neg(B' \leftrightarrow \neg A') = \neg((\neg B' \vee \neg A') \wedge (B' \vee A') = \neg(\neg B' \vee \neg A') \vee \neg(B' \vee A') = (B' \wedge A') \vee (\neg B' \wedge \neg A') = (A' \vee \neg B') \wedge (\neg A' \vee B')$.

Перепишем все высказывания в форме дизъюнктов:

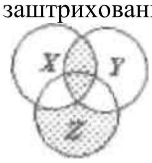
1. $\neg A \vee A'$.	8. $A' \vee B$.	(1, 3).
2. $\neg B \vee B'$.	9. $\neg B \vee A'$.	(2, 6).
3. $A \vee B$.	10. A'.	(8, 9).
4. $\neg A \vee \neg B'$.	11. $A \vee \neg A'$.	(3, 5).
5. $\neg B \vee \neg A'$.	12. $\neg A \vee \neg A'$.	(4, 7).
6. $A' \vee \neg B'$.	13. $\neg A'$.	(11, 12).
7. $\neg A' \vee B'$.	14. $\square$.	(10, 13).

Задание: докажите это заключение прямыми методами.

Упражнения.

Звездочками отмечены задания, для которых приведены решения или даны ответы.

1. Записать в символической форме следующие сложные высказывания:
   a) * если студент поздно ложится спать и на ночь пьет кофе, то утром у него плохое настроение и болит голова;
   b) * если деталь не стоит в плане и обеспечена заготовкой, то, если она срочная и план составлен правильно, - неверно, что деталь не стоит в плане;
   c) Иванов сдал экзамен и получил пять неравнозначно тому, что Иванов получил пять и сдал экзамен;
   d) если он едет, то он поезд, но это не значит, что если он стоит, то он не поезд;
   e) если ЭВМ исправна и мы не сделали ошибки, то мы сделали открытие тогда и только тогда, когда, если ЭВМ неисправна и мы не сделали ошибки, то мы все-таки что- то сделали.
2. * Какие из приведенных ниже высказываний являются отрицанием высказывания "Петя пошел в институт":
   - не Петя пошел в институт;
   - Петя не пошел в институт;
   - Петя пошел не в институт.
3. Каково значение истинности сложного высказывания "Речка движется и не движется"?
4. Придумать содержательное описание формул:
   a) $(A\leftrightarrow\neg B)\rightarrow A\wedge C$,
   b) $A\rightarrow(B\rightarrow(\neg A\vee C))$;
   c) $A|B\rightarrow(C\oplus D)$;
   d) $(A\rightarrow B)\leftrightarrow(A\rightarrow C)$.
5. Возможна ли ситуация, когда $A\vee\neg A \equiv$?.
6. Построить таблицы истинности для формул:
   a) $(A\rightarrow B)\vee(A\rightarrow B\wedge A)$;
   b) $(A\downarrow B)\rightarrow(A|C)$;
   c) $(A\wedge(B\rightarrow A))\rightarrow\neg A$;
   d) $(A\leftrightarrow B)\oplus C$;
   e) $(A|A)|(B|B)$;
   f) $A\rightarrow(B\rightarrow(C\rightarrow D))$.
7. * Выразить операции v и ^ через операции ¬ и →.
8. Выразить операции ¬, v и ^ через :
   a) штрих Шеффера;
   b) стрелку Пирса.
9. Построить тавтологию и противоречие с использованием трех элементарных высказываний A, B и C.
10. * Написать логическую формулу для заштрихованной области диаграммы Венна:



11. * Построить формулу от трех переменных:
    a) которая принимает такое же значение истинности, как и большинство переменных;
    b) истинную в том и только в том случае, когда совпадает значение ровно двух переменных.
12. Придумать тернарную, т.е. зависящую от трех переменных, логическую операцию.

13. Получить ДНФ для-формул:
    a) * ((x^y)v(x^y)vz)^(x^z);
    b) (xvy)^(¬xvz);
    c) * xvy;
    d) ((x^y)vz)^((x^z)vy);
    e) ¬(x^y)v¬(x^z);
    f) ¬(¬(x^y)^z);
    g) (x→y)^(y→x);
    h) ((x↔y)⊕z)|(x↓y);
14. Получить КНФ для формул:
    a) * (x^y)vz;
    b) * x^y;
    c) (x ^ y)v(¬x ^ z);
    d) ((x^y)vz)^(¬yv¬z);
15. Получить СДНФ для формул:
    a) * (xvy)^(¬xvz);
    b) * (x↓y) → (¬y|z);
    c) (x⊕y)|((xvy)↔(z → x));
    d) ¬(xvy)v(x^¬y);
    e) * xv¬xvy;
    f) x→y;
16. Получить СКНФ для формул:
    a) * (x^y)^(x^z);
    b) x↔y;
    c) * x^y;
    d) x^y^z;
    e) * x→(y→x);
    f) (x|y)→(x|z);
    g) (x|y)↔(x|z);
    h) (xvy)^(xvz)v(x^y);
    i) (xvy)^(¬x^z);
    j) (x↔y)⊕(x↔z);
17. Получить СДНФ, а затем перейти к СКНФ:
    a) (x→(y→z))→((x→¬z)→(x→¬y));
    b) * (x → y) → (¬y→¬x);
    c) (xvyvz) ^ (xv¬z)^¬y;
18. * Пусть задана функция f (сложное высказывание) от трех аргументов (элементарных высказываний) x, y и z и f(x, y, z)=x. Построить для данной функции СДНФ.
19. Получить СКНФ, а затем перейти к СДНФ:
    a) (x | y) → (z→¬y);
    b) (x→(y⊕z)) → z;
    c) (x↔y) | (y | z);
    d) * (x | y)^(x^y);
20. Получить МДНФ для формул:
    a) * ((x⊕y)↔z)→x;
    b) * (x⊕y) → ¬(xvy);
    c) ((A → B)↔(C↔D))v(¬(B → A)^(C ↔ D));
    d) * (¬Av¬BvC v ¬D)^(¬A v BvCv¬D);
    e) (x ↔ y)↔(x^z)
21. Методом резолюции показать выводимость заключения c из множества гипотез: {m, m→(nvp), n→c, p→c} ╞c.

ГЛАВА 2. ЯЗЫК ЛОГИКИ ПРЕДИКАТОВ ПЕРВОГО ПОРЯДКА.

2.1. Понятие предиката. Определения.

Исходные элементы в логике высказываний являются атомами. Из атомов строятся формулы. Каждый атом представляет собой повествовательное предложение, которое может быть истинным или ложным, но не одновременно и то и другое. Каждый атом рассматривается как единое целое. Структура атома и его состав не анализируется. Формулы могут выражать различные сложные мысли. Однако исчисление высказываний недостаточно для выражения, например, таких мыслей:

1. 1. "Каждый человек смертен. Сократ- человек. Следовательно, Сократ смертен".
2. 2. "Всякое положительное число есть натуральное число. Число 3 - положительное число. Следовательно, 3 есть натуральное число".

Каждое из приведенных рассуждений корректно. Однако, если мы введем, положим, для первого умозаключения обозначения

P: Каждый человек смертен,

Q: Сократ — человек,

R: Сократ - смертен,

то R не есть логическое следствие P и Q в рамках логики высказываний. Это объясняется тем, что исчисление высказываний ограничивается структурой предложений в терминах простых высказываний, а приведенные рассуждения требуют анализа структуры предложений в смысле связи субъекта и предиката, как это делается в грамматике. Хотя в логике слово "предикат" употребляется в более общем смысле, чем в грамматике, где оно выражает только то, что говорится о субъекте.

Пусть задано множество $V=\{v_1, v_2, \ldots, v_K, \ldots\}$, в котором v_1, v_2 и т.д. - какие-то объекты из этого множества. Обозначим любой предмет из этого множества через x и назовем x предметной переменной. Тогда высказывания об этих предметах обозначим через $P(v_1)$, $Q(v_1, v_2)$ и т.д., причем такие высказывания могут быть как истинными, так и ложными. Например, если $V=\{1, 2, 3, \ldots\}$, то высказывание "4 есть четное число" является истинным, а "7 есть четное число" - ложным. Если вместо конкретных чисел 4, 7 подставим предметную переменную x, то получим *предикат* их есть четное число", обозначаемый P(x). Таким образом, **предикат** на множестве V есть логическая функция, определенная на V, при фиксировании аргументов которой она превращается в высказывание со значениями {И, Л}.В приведенном примере имеем логическую функцию P(x), определенную на множестве V и принимающую значения: И, Л. В общем случае через $P(x_1, x_2, \ldots, x_n)$ обозначим n-местный предикат. Приписав значения переменным $x_1, x_2, \ldots, x_n$ из соответствующих областей определения, получим высказывание со значениями {И, Л}.

Определим *синтаксис* логики предикатов первого порядка.

- Алфавит
  1. Предметные переменные x, y, z, ... и предметные константы a, b, c, ...
  2. Функциональные константы или имена функций f_1, f_2, ...
  3. Предикатные константы или имена предикатов P^n, Q^K, R^j, ...
  4. Логические операторы (логические связки) $\neg$, ^(&), v, $\rightarrow$, $\leftrightarrow$, $\supset$.
  5. Символы кванторов $\forall$, $\exists$.
- Термы
  1. Всякая переменная или константа есть терм.
  2. Если $t_1, t_2, \ldots, t_n$ - термы и f_i^n - функциональная n-местная константа, то $f_i^n(t_1, t_2, \ldots, t_n)$ - терм. Здесь верхний индекс определяет количество аргументов. Если переменные терма t содержатся среди переменных списка $x_1, \ldots, x_s$. То пишем $t(x_1, \ldots, x_S)$.
  3. Никакие другие выражения не являются термами.

- •Атомы
  1. Если $t_1, \ldots, t_m$ - термы и P^m -m-местная предикатная константа, то $P^m(t_1, \ldots, t_m)$ - атом. При m=0 предикат P() обращается в высказывание P. Атомом является также равенство, т.е. выражение типа (s=t), где s и t - термы.
  2. Никаких других атомов нет.
- •Формулы:

  Формулы (или правильно построенные формулы ППФ) определяются следующим образом:
  1. Если $t_1, \ldots, t_K$ - термы, $\{x_1, \ldots, x_m\}$ - множество всех переменных, входящих в термы; P - предикатная переменная, то $P(t_1, \ldots, t_K)$ - элементарная формула (атом), а $\{x_1, \ldots, x_m\}$ — все её свободные переменные.
  2. Всякий атом есть формула.
  3. Если A и B - формулы, а x - свободная переменная, то ¬A, AvB, A^B, A→B, A↔B, ∀xA, ∃xA - формулы.
  4. Никаких других формул нет.

В пределах формализма логики предикатов первого порядка запрещено использование предикатов в качестве термов, навешивания кванторов на предикатный символ.

Предикат $P(t_1, t_2, \ldots, t_n)$ служит для выражения свойства объекта (при n=1) и отношений между объектами (при n≥2). Атомарная формула $P(a_1, a_2, \ldots, a_n)$ истинна в некотором "мире" (M), если между объектами в M имеется отношение, соответствующее P.

Связывание свободной переменной происходит либо квантификацией (взятием квантора по этой переменной) либо присвоением значения.

Например:

$\sum_{t=1}^{n} 2^i$: n - свободная переменная, i-связанная;

∀xP(x, y) : x - связанная переменная, y — свободная;

(x=a)P(x) : x - связанная переменная.

Если предметные переменные определены на конечной области, то операцию навешивания кванторов можно выразить через операции конъюнкции и дизъюнкции соответственно.

Пусть $x_1 \in \{a_1, a_2, \ldots, a_m\}$, тогда $\forall x_1 P(x_1, \ldots, x_n) =$ $P(a_1, x_1, \ldots, x_n)$^$P(a_2, x_1, \ldots, x_n)$^...^$P(a_m, x_1, \ldots, x_n)$,

$\exists x_1 P(x_1, x_2, \ldots, x_n) = P(a_1, x_1, \ldots, x_n)vP(a_2, x_1, \ldots, x_n)$v...v$P(a_m, x_1, \ldots, x_n)$; в частном случае

$\forall x P(x) = P(a_1)$^$P(a_2)$^...^$P(a_m)$,

$\exists x P(x) = P(a_1)$v$P(a_2)$v...v$P(a_m)$.

Говорят, что квантор связывает соответствующую переменную. Предикат (формула) называется **замкнутым**, если связаны все переменные (если нет свободных, т.е. несвязанных переменных).

Примеры.

1. Термами являются:
   a) переменные: x, фамилия, число, буква;
   b) константы: C, Дейкстра, 94, ъ;
   c) функции max(x, y, z), ∑(x, y) или более традиционное представление - (x+y).
2. Термами не являются выражения:
   a) P(x) - так как P - символ, зарезервированный для предиката;
   b) (x>0) - так как это отношение (т.е. с точки зрения логики -одноместный предикат);
   c) x:=a - специфическая операция в программировании.
3. Формулами являются:
   a) P(x)=(x>0), выше(x, y) или в более традиционном представлении (x выше y);
   b) формулы, полученные из атомарных ¬P(x), ∀xP(x), P(x)→Q(x, y), ∀x∃yQ(x, y)v¬zR(x, z), (x выше y)^(y выше z)→(x выше z).
4. 4. Формулами не являются:
   a) (a+b) - поскольку это терм;
   b) (a+b)^(c-d) - здесь два терма, соединенные логическими символами;

c) $\forall P \exists x P(x)$ - поскольку здесь квантор $\forall$ навешен на символ предиката (это выражение могло бы служить примером формулы в языке второго порядка).

Из интуитивного понимания квантификации вытекает, что имеется связь между вхождениями переменной, содержащейся в квантификации. Так, в формуле $\forall x(P(x)\rightarrow Q(x))$ переменная x связана. Связь вводится первым вхождением, которое следует за квантором общности.

Область действия некоторой квантификации есть формула, к которой применяется эта квантификация. Переменная x в кванторах $\forall x$ и $\exists x$ называется квантифицированной. Каждое вхождение переменной x в область действия этой квантификации является **связанным**. Вхождение переменной x **свободное**, если оно не является ни квантифицированным, ни связанным.

В формулах $\exists xA$ и $\forall xA$ область действия x есть A. Если x и y - переменные, то области действия x и y либо не пересекаются, либо одна включает другую. Рассмотрим примеры:

$\forall x[P(x, a)\rightarrow \exists x Q(x)]$,

$\forall x P(x, a)\rightarrow \exists x Q(x)$,

$\forall x P(x, a)\rightarrow Q(x)$.

В первом случае две квантификации по одной переменной перекрываются. Такой тип формул исключим из рассмотрения, придерживаясь правила введения кванторов: $\exists xA$ и $\forall xA$ - формулы только в том случае, если x - переменная и A - формула, не содержащая связанных вхождений x.

Во втором случае две связанные переменные случайно получили одинаковое имя.

В третьем случае и связанная переменная и свободная получили одно имя. Следовало бы сразу переименовать, например, связанную переменную: $\forall y P(y, a)\rightarrow Q(x)$.

Пример.

2.2. Интерпретации. Модели.

Формулы исчисления предикатов (как и формулы исчисления высказываний) могут быть интерпретированы, т.е. могут получить некоторое *значение истинности*.

Однако составные части формул исчисления предикатов являются не только формулами, но и термами. Поэтому необходимо интерпретировать еще и термы. Интуитивно терм означает объект. Тогда интерпретация должна указывать множество объектов, называемое **областью интерпретации**.

Интерпретация /есть тройка (S, Ic, I_v), где:

- S - непустое множество (область интерпретации);
- Ic - функция, сопоставляющая каждой n-местной функциональной константе f функцию Ic(f) из S^n в S. Она также каждой m-местной предикатной константе P ставит в соответствие функцию Ic(P) из S^m в {**И**, **Л**}.
- I_v - функция, сопоставляющая каждой переменной какой-то элемент из S.

Необходимо ввести еще обозначение, которое послужит для интерпретации двух видов квантифицированных формул. Если I - некоторая интерпретация с областью S_I, x - переменная и d - элемент из S_I то Ix/d означает такую интерпретацию J, что S_j=S_I, Jc=Ic, J_v(x)=d, J_v(y)=I_v(y) для всех переменных y, отличных от x. Теперь для каждой интерпретации I=(S, Ic, I_v) можно задать такие правила интерпретации, которые каждой формуле A сопоставляют значение истинности I(A) и каждому терму t сопоставляют элемент I(t) из S. Эти правила интерпретации образуют *семантику* языка логики предикатов первого порядка.

Семантика.

- Если x - переменная, то I(x)=I_v(x) (*по определению*).
- Если f - n-местная функциональная константа и t_1, …, t_n -термы, то I(f(t_1, …, t_n)) = Ic(f)(I(t_1), ..., I(t_n)) (*по определению*).
- Если P - m-местная предикатная константа и t_1, …, t_n - термы, то I(P(t_1, …, t_n) = Ic(P)(I(t_1), …, I(t_n)) (*по определению*).
- Если s и t - термы, то I(s=t) есть **И**, если I(s)=I(t), в противном случае — **Л**.
- Если A и B - формулы, то ¬A, (A^B), (AvB), (A→B) и (A↔B) интерпретируются так же, как и в исчислении высказываний.
- Если A - формула и x - переменная, то I(∀xA) есть **И**, если Ix/d(A) есть **И** для всех элементов d из S.
- Если A - формула и x - переменная, то I(∃xA) есть **И**, если Ix/d(A) есть **И** хотя бы для одного d из S.

Формула A исчисления предикатов первого порядка называется истинной при интерпретации f, если I(A)=**И**.

Говорят, что терм t **свободен** для переменной x в формуле P, если ни x, ни произвольная переменная из t не квантифицированы в P. При этом через Px/t обозначается формула, полученная из P путем одновременной замены всех вхождений x на t. Формула называется **замкнутой**, если в ней нет несвязанных (свободных) переменных.

Замкнутая формула, истинная в любой интерпретации называется **общезначимой**. Замкнутая формула, ложная при всех интерпретациях, называется **невыполнимой**, а истинная при некоторых интерпретациях - **выполнимой**.

Мы определили семантику формул логики предикатов первого порядка, основанную на понятии интерпретации. Теперь определим семантику, опираясь на понятии *модели*. Эти понятия отличаются лишь видами используемых словарей и обозначениями.

Моделью *M* логики предикатов первого порядка называется пара (S,V), где S - область интерпретации и V - функция, совпадающая с функцией интерпретации I. Роль функции V состоит в том, чтобы интерпретировать функциональные и предикатные константы языка в терминах элементов S.

Пусть {t} - список всех свободных переменных, входящих в терм t, а g - функция, приписывающая всякой индивидной переменной из t значение из S. Индуктивным определением введем понятие терма t в модели M при присваивании g, обозначив это понятие U(t, g). Функция g соответствует функции I_v интерпретации I. Функция g не является частью модели M.

Семантика формул языка логики предикатов первого порядка описывается с помощью обозначения

$$M \models_g A,$$

которая означает "A истинна в модели M для назначения g".

Тогда для любой формулы A имеем эквивалентные обозначения:

$M \models_g A \Leftrightarrow U(A, g)=1, \qquad I(A)=1 \Leftrightarrow U(A, g)=1.$

Теперь мы можем сделать следующие определения:

- Если терм t есть константа a_i, то $U(t, g)=I(a_i)$, т.е. индивидной константе сопоставляется элемент множества S: $I(a_i) \in S$.
- Если терм t есть переменная x_i, то в данной модели $U(t, g)=g(x)$.
- Если f - n-местная функциональная константа и $t_1, \ldots, t_n$ -термы, то $U(f(t_1, \ldots, t_n)) = V(f)(U(t_1, g), \ldots, U(t_n, g)))$.
- Если P n-местная предикатная константа и $t_1, \ldots, t_n$ – термы, то $U(f(t_1, \ldots, t_n)) =$
- $V(f)(U(t_1, g), \ldots, U(t_n, g)))$.

В описанной семантике можно дать определение понятия выполнимости.

$M \models_g P(t_1, \ldots, t_n)$ тогда и только тогда, когда значение истинности $V(P(t_1, \ldots, t_n))$ есть **И** при данной функции приписывания g.

- $M \models_g \neg A$ тогда и только тогда, когда $M \not\models_g A$.
- $M \models_g A \wedge B$ тогда и только тогда, когда $M \models_g A$ и $M \models_g B$.
- Аналогично для формул AvB, $A \rightarrow B$ и $A \leftrightarrow B$.
- $M \models_g \forall x A$ тогда и только тогда, когда $M \models_g A$ для всех присваиваний g, которые отличаются от присваивания g только сопоставлением значения переменной x.

Формула A общезначима тогда и только тогда, когда $M \models_g A$ для всех моделей M и всех присваиваний g.

Пример. Рассмотрим формулу:

$\forall x(P(x) \rightarrow Q(f(x), a))$.

В этой формуле имеем: индивидную константу a, одноместную функциональную константу f, унарную предикатную константу P (или просто предикат) и бинарный предикат Q. Задана следующая интерпретация I:

S={1, 2};

I(a): {1, 2}; примем a=1 для всех $x \in S$.

$I_v(a)$: {1, 2};

$I(f(x)) = Ic(f)(I_v(x)) = \{2(1), 1(2)\}$;

$I(P(x)) = Ic(P)(I_v(x)) = \{Л(1), И(2)\}$;

$I(Q(f(x))) = Ic(Q)(Ic(f)(I_v(x)), I(a)) = \{И(1(2), 1), И(1(2), 2), Л(2(1), 1), И(2(1), 2)\}$.

При x=1

$P(x) \rightarrow Q(f(x), a) = P(1) \rightarrow Q(f(1), a) = P(1) \rightarrow Q(2, 1) = Л \rightarrow Л = И$.

При x=2

$P(x) \rightarrow Q(f(x), a) = P(2) \rightarrow Q(f(2), a) = P(2) \rightarrow Q(1, 1) = И \rightarrow И = И$.

Таким образом, для всех x из S формула $P(x) \rightarrow Q(f(x), a)$ истинна, следовательно, она истинна в интерпретации I.

Пример. В интерпретации из предыдущего примера оценим выполнилось формулы:

$\exists x(P(f(x)) \wedge Q(x, f(a)))$.

При x=1

$P(f(x)) \wedge Q(x, f(a)) = P(f(1)) \wedge Q(1, f(a)) = P(2) \wedge Q(1, f(1)) = P(2) \wedge Q(1, 2) = И \wedge И = И$.

При x=2

$P(f(x)) \wedge Q(x, f(a)) = P(f(2)) \wedge Q(2, f(a)) = P(1) \wedge Q(2, f(1)) = P(1) \wedge Q(2, 2) = Л \wedge И = Л$.

Поскольку для $x \in S$, т.е. x=1, формула $P(f(x)) \wedge Q(x, f(a))$ истинна, то формула $\exists x(P(f(x)) \wedge Q(x, f(a)))$ истинна в интерпретации I.

Правило вывода в логике предикатов определяется как правило логического следования. Формула B логически следует из формул $A_1, A_2, \ldots, A_n$ ($A_1, A_2, \ldots, A_n \vdash B$) тогда и только тогда, когда всякая интерпретация, удовлетворяющая $A_1 \wedge A_2 \wedge \ldots \wedge A_n$, удовлетворяет также и B, т.е. из истинности посылок $A_1, A_2, \ldots, A_n$ следует истинность заключения B. В соответствии с определением формальной дедуктивной системы, множество выводимых формул образовано формулами, которые логически следуют из аксиом или аксиом и ранее введенных формул.

Следующие две теоремы позволяют определить справедливость логического следования $A_1, A_2, \ldots, A_n \vdash B$ из общезначимости формулы $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$ или противоречивости формулы $A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \neg B$.

Теорема дедукции. *Формула B является логическим следствием* $A_1, A_2, \ldots, A_n$ *тогда и только тогда, когда формула* $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$ *общезначима.*

Доказательство. Необходимость. Пусть $A_1, A_2, \ldots, A_n \vdash B$, тогда $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$ общезначима.

Обозначим через $I\{i_1, i_2, \ldots, i_K, \ldots\}$ множество интерпретаций упорядоченных таким образом, что первые k позиций занимают интерпретации, в которых формула $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ истинна, а затем перечислены интерпретации, в которых формула $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ ложна. Из определения логического следования вытекает, что возможны два случая, представленных в табл.2.1 и 2.2.

Таблица 2.1.

I	$A_1 \wedge A_2 \wedge \ldots \wedge A_n$	B	$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$
i_1	И	И	И
i_2	И	И	И
…	...	...	...
I_K	И	И	И
...	Л	Л	И
…	Л	Л	И
…	…	…	…
…	Л	Л	И

Таблица 2.2.

I	$A_1 \wedge A_2 \wedge \ldots \wedge A_n$	B	$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$
i_1	И	…	И
i_2	И	...	И
...	...	…	…
I_K	И	…	И
…	И	…	И
…	И	…	И
…	…	…	…
…	Л	…	И

Достаточность. Пусть $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$ общезначима, тогда справедливо $A_1, A_2, \ldots, A_n \vdash B$.

Таблица 2.3

A	B	A→B
И	И	И
И	Л	Л
Л	И	И
Л	Л	И

Из таблицы истинности формулы A→B (табл.2.3) следует, что общезначимость формулы $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$ исключает существование интерпретаций, в которых $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ истинна, а B - ложна. Следовательно, из истинности формулы $A_1 \wedge \ldots \wedge A_n$ следует истинность B, т.е. $A_1, A_2, \ldots, A_n \vdash B$.

Теорема противоречивости. *Формула B является логическим следствием* $A_1, A_2, \ldots, A_n$ тогда и только тогда, когда формула $A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \neg B$ противоречива.

Из теоремы следует, что для доказательства выводимости формулы B из формул $A_1, A_2, \ldots, A_n$ достаточно доказать противоречивость формулы $A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \neg B$. Это утверждение лежит в основе метода *резолюции* (опровержения). Для применения этого метода необходимо представить формулы в стандартизованном виде, который представляет собой конъюнкцию дизъюнкций. Приведение формул к стандартной форме рассмотрены в последующих разделах.

Пример. Пусть заданы формулы:

F_1: $\forall x(P(x) \rightarrow Q(x))$;

F_2: $P(a)$.

Показать, что $Q(a)$ является логическим следствием F_1 и F_2.

Пусть задана некоторая интерпретация I, в которой формула $\forall x(P(x)\rightarrow Q(x))$ истинна. Будем считать, что в формуле F_2 P(a) истинна. Положим, что Q(a) ложна в принятой интерпретации I, тогда импликация $P(a)\rightarrow Q(a)\equiv\neg P(a)vQ(a)$ ложна. Следовательно, ложна в I и формула $\forall x(P(x)\rightarrow Q(x))$, а это невозможно в силу исходного предположения. Тогда следует, что Q(a) истинна в любой интерпретации, в которой формула $\forall x(P(x)\rightarrow Q(x))\wedge P(a)$ истинна. Следовательно, Q(a) является следствием F_1 и F_2

Приведем еще несколько примеров, иллюстрирующих рассмотренный материал.

1. Общезначимые формулы:
   $\forall xP(x)\rightarrow\forall yP(y)$ - тавтология;
   $\forall xP(x)\rightarrow\exists yP(y)$ - при любой непустой области интерпретации из справедливости для всех элементов следует справедливость для отдельных;
   $P(a)v\neg P(a)$ - тавтология;
2. Невыполнимые формулы:
   $\forall xP(x)\rightarrow\neg\forall yP(y)$ - противоречие;
   $\forall xP(x)\rightarrow\neg P(a)$ - из справедливости для всех следует несправедливость дня отдельного элемента, что также является противоречием.
3. Выполнимые формулы:
   $P(a)\rightarrow\neg\exists xP(x)$ - формула истинна в интерпретациях, в которых P(a) ложна. Например, если P(x) интерпретируется как " x - отличник" и в рассматриваемом множестве D студентов группы Иванов не является отличником, тогда формула $P(\text{Иванов})\rightarrow\neg\exists xP(x)$ - истинна;
   $\forall x(P(x)v\neg Q(x))$ - формула, истинная в интерпретациях, в которых для каждого элемента истинно P или ложно Q;
   $\exists xP(x)$ - формула истинная в интерпретациях, в которых хотя бы для одного элемента истинно P.
   Напомним, интерпретация называется моделью для данного множества формул, если все формулы рассматриваемого множества истинны (выполнимы) в данной интерпретации.
4.
   a) Моделью для формул
      $\forall x\wedge\neg P(x, x)$, $\forall x, y, z(P(x, y)\wedge P(y, z)\rightarrow P(x, z))$, $\forall x\exists yP(x, y)$
      является множество действительных чисел с заданным на нем отношением меньше, т.е. P(x, y) - «x меньше j».
   b) Моделью для формул
      $\forall x(O(x)\wedge P(x)\rightarrow \Pi(x))$, $\forall x(\neg O(x)v\neg P(x)\rightarrow\neg\Pi(x))$, $\exists x\neg\Pi(x)$,
      где O(x) - "x учится отлично", P(x) - « x занимается общественной работой » и П(x) - "x получает повышенную стипендию", могут служить реально существующие принципы, используемые при назначении стипендии.
   c) в) Моделью для формулы
      $$\forall\varepsilon\exists\delta\forall x(R(\varepsilon)\wedge P(x,\delta)\rightarrow Q(f(x),\beta,\varepsilon))$$
      является следующая интерпретация:
      S- множество действительных чисел;
      $P(x, \alpha, \delta)$: $|x-\alpha|<\delta$;
      $Q(f(x), \beta, \varepsilon)$: $|f(x)-\beta|<\varepsilon$;
      $R(\varepsilon)$: $\varepsilon>0$;
      f(x) - произвольная, фиксированная функция, заданная на отрезке $[a, \beta]$; x, δ, ε - индивидные переменные из S; α, β - константы из S, $\alpha\in[a,b]$. В такой интерпретации приведенная формула соответствует утверждению о том, что число β является пределом функции f(x) при $x\rightarrow\alpha$.

2.3. Правила вывода в логике предикатов первого порядка.

В процессе решения задачи мы используем различные способы рассуждений, которые приводят к поставленной цели. Мы можем декомпозировать задачу на подзадачи, каждая из которых имеет свое частичное решение, или вывод, и при этом этот частичный вывод становится основой для решения следующей подзадачи и т.д. В любом.случае началом рассуждений является гипотеза или совокупность гипотез, из которых на базе некоторых правил (законов) делается заключение. В самом начале мы уже отмечали, что заключения типа "потому что" или "если... то" являются примерами рассуждений. Интуитивно можно определить процесс решения задачи как вывод из гипотез и аксиом. И эта идея получила отражение в тех рассуждениях, которые приведены в предыдущем параграфе и. в частности, в теореме дедукции. Теперь дадим более строгое определение вывода.

Выводом из множества гипотез (посылок, допущений) назовется непустая конечная последовательность формул, в которой каждая формула есть одна из гипотез, или формула, полученная из предшествующих формул последовательности по одному из правил вывода, или теорема; вывод последней формулы >той последовательности называется заключением из исходного множества гипотез.

Доказательством выводимости называется непустая конечная последовательность выводимостей, в которой каждая выводимость или является непосредственно обоснованной, или же получена из предшествующих выводимостей по одному из правил:

- правило дедукции $(Г, A \vdash B) \vdash (Г \vdash A \to B)$;
- правило доказательства от противного $(Г, \neg A \vdash B \wedge \neg B) \vdash (Г \vdash A)$;
- правило сведения к абсурду $(Г, \neg A \vdash B \wedge \neg B) \vdash (Г \vdash \neg A)$.

Эти правила называются **правшами второго рода**, в отличие от правил В1 - В12.(гл1, §1.5), которые называются **правилами первого рода**. В приведенных правилах Г - множество гипотез (формул), возможно непустое, А, В - формулы.

Выводимость, для которой есть доказательство, называется **обоснованной**.

В приведенном определении вывода осталось нераскрытым понятие теоремы. Существует несколько равнозначных определений понягия теоремы. Приведем два из них:

Если В есть логическое следствие формул $A_1, \ldots, A_n$, то формула $((A_1, \ldots, A_n) \to B)$ называется **теоремой**, а В - **заключение** теоремы.

Формула А называется **теоремой**, если и только если существует обоснованная выводимость $Г \vdash A$, для которой Г пусто.

Например, пусть требуется обосновать выводимость формулы $\vdash (c \to d) \to (\neg d \to \neg c)$, т.е. установить, что приведенная формула является теоремой. Естественно, что доказательство должно опираться на посылки (гипотезы). Правомочен вопрос, как определить в поставленной задаче гипотезы. Возможны два способа выделения гипотез:

- если главный знак (логическая связка) формулы не является конъюнкцией или эквивалентностью, то можно взять в качестве единственной гипотезы отрицание этой формулы. Для приведенного примера
  $\neg((c \to d) \to (\neg d \to \neg c))$;
- если главным знаком формулы является импликация, то в качестве гипотезы взять антецедент (посылку) этой формулы (в примере $c \to d$);
- если консеквент (следствие) формулы имеет главным знаком импликацию, то в качестве второй гипотезы взять антецедент консеквента и т.д. Из полученных гипотез следует вывести консеквент последнего консеквента. Если это не удается, то взять в качестве допущения (дополнительной гипотезы) отрицание последнего консеквента и вывести противоречие.
- Если главным знаком формулы является конъюнкция, то сначала доказываются в качестве теорем члены конъюнкции. Например, доказать формулу $A \wedge B$:
  1. А - теорема;
  2. В - теорема;
  3. $A \wedge B$ - из 1 и 2 по П1 (см. ниже).
  4. $\vdash A \wedge B$ по определению вывода на основе 1 — 3.

Для приведенного выше примера будем иметь две гипотезы:

$c \to d$ и $\neg d$.

Заметим сразу, что правила (первого рода), приведенные в §1.5 правомерны и для логики предикатов первого порядка, только надо иметь в виду, что литеры в этих правилах представляют предикатные формулы. Специфика же для логики предикатов первого порядка привносит дополнительные правила, связанные с квантификацией переменных. Не будем вы-. писывать эти формулы вновь, но будем иметь в виду, что при анализе предикатных формул ссылка на правило с меткой Пi соответствует правилу Вi. Здесь же мы выпишем правила, специфические для логики предикатов первого порядка:

П14. Отрицание квантора существования: $\neg\exists x A(x) \vdash \forall x \neg A(x)$

П15. Удаление квантора общности: $\forall x A(x) \vdash A(t)$, где $A(t)$ - результат правильной подстановки терма t вместо x;

П16. Введение квантора существования: $A(t) \vdash \exists x A(x)$, где $A(t)$ - результат правильной подстановки терма t вмесго x;

П17. Введение квантора общности: $A(x) \vdash \forall y A(y)$, где $A(y)$ - результат правильной подстановки переменной y вместо x.

П18. Удаление квантора существования: $\exists y A(y) \vdash A(x)$, где $A(x)$ - результат правильной подстановки переменной x вместо y в $A(y)$;

П19. Правило дедукции: $(\Gamma, A \vdash B) \vdash (\Gamma \vdash A \to B)$,

П20. Доказательство от противного: $(\Gamma, \neg A \vdash B \wedge \neg B) \vdash (\Gamma \vdash A)$;

П21. Сведение к абсурду: $(\Gamma, \neg A \vdash B \wedge \neg B) \vdash (\Gamma \vdash \neg A)$.

Правила П1- П18 являются правилами первого рода, правила П19 - П21 - правила второго рода.

Приведем пример использования правил для доказательства выводимости. Для начала завершим приведенный пример. Итак, доказать (вывести) теорему

$(c \to d) \to (\neg d \to \neg c)$.

1. $c \to d$ - гипотеза;
2. $\neg d$ - гипотеза;
3. $\neg c$ - из 1 и 2 по П7.

Получили множество формул $\{c \to d, \neg d\}$, из которого последовал вывод $\neg c$, т.е.

4. $\{c \to d, \neg d\} \vdash \neg c$;
5. $(c \to d \vdash (\neg d \to \neg c)$ из 4 по П19;
6. $\vdash (c \to d) \to (\neg d \to \neg c)$ из 5 по П19.

Следует сделать существенное замечание по поводу применения правил П13 - П18. В прямом выводе необходимо отслеживать правильность подстановок переменных, связанных и свободных. Это в значительной степени усложняет доказательство выводимости и, главное, делает проблематичным использование для машинной реализации. В то время как описанный в следующем параграфе способ преобразования формул к приведенной нормальной форме и анализ множества формул на выводимость с помощью метода резолюции лишен этого недостатка, а убедительным примером реализуемости вывода на основе метода резолюции является язык логического программирования Пролог.

2.4. Получение дизъюнктов.

Принцип резолюции работает со специфическим подмножеством языка первого порядка, который будем называть языком дизъюнктов. В этом разделе дадим конструктивное определение дизъюнкта (т.е. фактически опишем, как получать дизъюнкты).

Формула вида $Q_1x_1...Q_nx_n$ (M), где каждые Q_ix_i есть либо $\forall x_i$, либо $\exists x_i$, а M -формула, не содержащая кванторов, называется формулой в **предваренной нормальной форме**.

При преобразовании формулы в предваренную нормальную форму можно воспользоваться следующими основными равносильностями:

$\neg\forall xP(x) \equiv \exists x\neg P(x)$, (2.1)

$\neg\exists xP(x) \equiv \forall x\neg P(x)$, (2.2)

$\neg\forall x\neg P(x) \equiv \exists xP(x)$, (2.3)

$\neg\exists x\neg P(x) \equiv \forall xP(x)$, (2.4)

Пусть R - формула, не содержащая переменной x, тогда

$\forall xR \equiv R$, (2.5)

$\exists xR \equiv R$, (2.6)

$\forall xP(x)vR \equiv \forall x(P(x)vR)$, (2.7)

$\forall xP(x){\wedge}R \equiv \forall x(P(x){\wedge}R)$, (2.8)

$\exists xP(x)vR \equiv \exists x(P(x)vR)$, (2.9)

$\exists xP(x){\wedge}R \equiv \exists x(P(x){\wedge}R)$. (2.10)

Законы (2.7), (2.9) и (2.8), (2.10) можно обобщить законами:

$(Qx)P(x)vR \equiv (Qx)(P(x)vR)$, (2.9a)

$(Qx)P(x){\wedge}R \equiv (Qx)(P(x){\wedge}R)$, (2.10a)

$\forall xP(x){\wedge}\forall xR(x) \equiv \forall x(P(x){\wedge}R(x))$, (2.11)

$\exists xP(x)v\exists xR(x) \equiv \exists x(P(x)vR(x))$, (2.12)

$\forall xP(x) \equiv \forall yP(y)$, (2.13)

$\exists xP(x) \equiv \exists yP(y)$. (2.14)

Формулы (13) и (14) имеют место при условии, что x и y принимают свои значения из одной и той же области.

$\forall xP(x)v\forall xR(x) \not\equiv \forall x(P(x)vR(x))$, (2.15)

$\exists xP(x){\wedge}\exists xR(x) \not\equiv \exists x(P(x){\wedge}R(x))$, (2.16)

Поскольку каждая связанная переменная в формуле может рассматриваться как место для подстановки, то следует в этих формулах одно из вхождений x переименовать, например, на z. Пусть выбрана переменная z, которая не встречается в P(x), тогда формулы (2.15) и (2.16) принимают вид:

$\forall xP(x)v\forall xR(x) \equiv \forall x\forall z(P(x)vR(z))$, (2.17)

$\exists xP(x){\wedge}\exists xR(x) \equiv \exists x\exists z(P(x){\wedge}R(z))$, (2.18)

В общем случае имеем $Q_1x_1...Q_nx_n$

$(Q_1x)P(x)v(Q_2x)R(x) \equiv (Q_1x)(Q_2z)(P(x)vR(z))$, (2.17a)

$(Q_1x)P(x){\wedge}(Q_2x)R(x) \equiv (Q_1x)(Q_2z)(P(x){\wedge}R(z))$, (2.18a)

$\exists x\forall yP(x, y) \not\equiv \exists y\forall xP(x, y)$, (2.19)

$\exists x\forall yP(x, y) \not\equiv \forall y\exists xP(x, y)$. (2.20)

Далее, та часть формул, которая не имеет кванторов, приводится к конъюнктивной нормальной форме с использованием алгебраических преобразований для операций ¬, ^(&), v, →, ↔.

Приведем алгоритм преобразования формул в предваренную нормальную форму.

Шаг 1. Исключаем логические связки → и ↔, используя законы

$P\leftrightarrow R \equiv (P\to R){\wedge}(R\to P)$,

$P\to R \equiv \neg PvR$.

Шаг 2. Используем законы

- двойного отрицания: $\neg(\neg P) \equiv P$;
- законы де Моргана:
  $\neg(PvR) \equiv \neg P \wedge \neg R$,
  $\neg(P \wedge R) \equiv \neg P v \neg R$;
- для внесения отрицания внутрь формулы используем:
  $\neg\forall xP(x) \equiv \exists x\neg P(x)$,
  $\neg\exists xP(x) \equiv \wedge\forall\neg P(x)$.

Шаг 3. Переименование связанных переменных.

Шаг 4. Используем законы (2.9а), (2.10а), (2.11), (2.12), (2.17а) и (2.18а).

Шаг 5. Используем закон дистрибутивности, заменяя формулы вида
$(A \wedge B)vC$ на $(AvC) \wedge (BvC)$;
$Av(B \wedge C)$ на $(AvB) \wedge (AvC)$.

И, наконец, чтобы избавиться от кванторов, используется процедура **сколемизации**, которая состоит в следующем: кванторы просматриваются слева направо до первого квантора существования $\exists$, при этом

- если в кванторной приставке $Q_1x_1...Q_nx_n$ квантор $\exists x_i$ не имеет левее себя кванторов $\forall x_j$, то везде в формуле М переменная x_i заменяется на некоторую неопределенную константу;
- если в кванторной приставке квантор $\exists x_i$ стоит правее кванторов $\forall x_{j1}, \ldots, \forall x_{jK}$ (k<n), то везде в формуле М переменная x_i заменяется на некоторую неопределенную функцию $f_i^C(x_{j1}, \ldots, x_{jK})$,
- после исключения всех кванторов существования кванторы общности просто отбрасываются. Считается, что все оставшиеся переменные связаны кванторами общности, поэтому их отбрасывание не влечет неоднозначностей в понимании формул.

Элементарные дизъюнкции, полученные в результате выполнения описанных выше процедур, называются **дизъюнктами**.

Примеры.

1. Пусть задано множество X целых чисел $x, y \in X$ и введем предикат P(y, y), устанавливающий отношение "больше" между двумя числами. Тогда на заданном множестве для $\forall x$ существует y такое, что y>x, т.е. справедлива истинность предиката P(y, x): $\forall x\exists yP(y, x)$.

В соответствии с приведенным выше алгоритмом сколемизации имеем: $P(f^C(x),)$. Тогда на основании веденного в примере определения предиката сколемовская функция в частности может иметь вид: $f^C(x)=x+1$.

2. Получить дизъюнкты для формулы

$\exists x(\exists yP(x, y)v\exists zQ(x, z)\rightarrow\exists vR(x,v))$.

Избавимся от импликации:

$\exists x(\neg(\exists yP(x, y)v\exists zQ(x, z))v\exists vR(x, v)) = \exists x(\neg\exists yP(x, y)\wedge\neg\exists zQ(x, z)v\exists vR(x, v) =$
$\exists x(\forall y\neg P(x, y)\wedge\forall z\neg Q(x, z)v\exists vR(x, v))$

Вынесем кванторы в начало формулы:

$\exists x\forall y, z\exists v(\neg P(x, y)\wedge\neg Q(x, z)vR(x, v))$.

Отметим, что это один из множества допустимых вариантов вынесения кванторов. Здесь возможно получение кванторной приставки $\exists x\forall z, y\exists v$ и даже $\exists x, v\forall y, z$

Полученную на предыдущем шаге формулу приводим к конъюнктивной нормальной форме:

$\exists x\forall y, z\exists v((\neg P(x, y)vR(x, v)) \wedge (\neg Q(x, z)vR(x, v)))$.

И, наконец, выполняем сколемизацию:

$(\neg P(a^C, y)vR(a^C, f^C(y, z)))\wedge(\neg Q(a^C, z)vR(a^C, f^C(y, z)))$. В результате получаем два дизъюнкта, которые обычно записываются через запятую:

$(\neg P(a^C, y)vR(a^C, f^C(y, z)))$ и $(\neg Q(a^C, z)vR(a^C, f^C(y, z)))$.

2. Получить дизъюнкты для формулы:

$\forall x\exists yP(x, y)\wedge\neg\exists y\forall zR(y, z) = \forall x\exists yP(x, y)\wedge\forall y\exists z\neg R(y, z) =$
$\forall x\exists yP(x, y)\wedge\forall v\exists z\neg R(v, z) = \forall x\exists y\forall v\exists z(P(x, y)\wedge\neg R(v, z)) = P(x, f_1^C(x))\wedge\neg R(v, f_2^C(x, y))$.

В результате имеем два дизъюнкта

$P(x, f_1^C(x))$ и $R(v, f_2^C(x, y))$.

2.5. Теорема Эрбрана.

Если формула в языке первого порядка невыполнима, т.е. не имеет интерпретации, то невыполнимым является и множество дизъюнктов, полученное из этой формулы (т.е. нет интерпретации, которая удовлетворяла бы одновременно всем дизъюнктам множества). Справедливо также и обратное утверждение.

Практически нереально рассматривать все возможные интерпретации на всех возможных множествах. Тем не менее можно ограничить, не уменьшая общности рассуждений, как перечень рассматриваемых множеств, так и перечень возможных интерпретаций на этих множествах. Более того, достаточно зафиксировать одно такое множество H для данного множества дизъюнктов S (кратко H(S)), что S невыполнимо тогда и только тогда, когда S ложно при всех интерпретациях на этом множестве. Такое множество H(S) называют **универсумом** Эрбрана для S и определяют следующим образом:

1. Множество всех констант из S принадлежит или, если в H(S) нет констант, принимается, что H(S) принадлежит некая произвольная константа a.
2. Если термы принадлежат H(S), то H(S) принадлежит также $f(t_1, \ldots, t_n)$, гдеf -функциональный символ, упомянутый в S.

Примеры.

1. Пусть S={P(x), R(x, y), Q(y)}, тогда H(S) ={a}.
2. Пусть S={P(a), R(x, y), Q(y)}, тогда H(S) ={a}.
3. Пусть S={P(a), R(x, b), Q(c)}, тогда H(S) ={a,b,c}.
4. Пусть S={P(a), R(f(x), b), Q(φ(x, z))},Тогда

H(S)={a, b, f(a), f(b), φ(a, a), φ(a, b), …, f(f(a)), …, f(φ(a, b)), …, φ(f(φ(a, a)), φ(f(a), φ(b, a))), …}.

Наличие хотя бы одной функции делает универсум Эрбрана бесконечным (счетным).

Вне зависимости от интерпретации никаких Термов для S, выходящих за рамки универсума Эрбрана, быть не может.

Фундаментальной (основной) конкретизацией (Прим.: В некоторых источниках [35] термин instance в русском переводе дан в значении пример. Более точным по смыслу, на наш взгляд, является значение *конкретизация*.) дизъюнкта C называется дизъюнкт, полученный замещением переменных в C членами универсума Эрбрана H(S) таким образом, что все вхождения одной и той же переменной в C заменяется на один и тот же терм. Аналогично определяется фундаментальный пример предиката.

Базой Эрбрана для всех множеств дизъюнктов S называется множество всех фундаментальных конкретизации предикатов из S.

Интерпретация I называется **H-интерпретацией** (интерпретацией Эрбрана) тогда и только тогда, когда она удовлетворяет следующим условиям:

1. I отображает все константы из S в себя.
2. Пусть f есть n-местный функциональный символ и $t_1, \ldots, t_n$ - элементы H(S), тогда. I: $(t_1, \ldots, t_n) \rightarrow f(t_1, \ldots, t_n)$, т.е. I отображает кортеж термов в соответствующее значение функции f. H-интерпретация относится только к термам и не накладывает никаких ограничений на предикаты (атомы).

Пусть A = $\{A_1, \ldots, A_n, ...\}$ является базой Эрбрана для S. Тогда H-интерпретация может быть представлена множеством $I\{m_1, \ldots, m_n, ...\}$, где m_j есть A_j или $\neg A_j$. При этом, если m_j есть A_j, то A_j приписывается значение "истинно", иначе A_j приписывается значение "ложно". Существует теория, утверждающая, что множество дизъюнктов невыполнимо тогда и только тогда, когда S ложно при всех H-интерпретациях.

Таким образом, мы ограничили исходное множество универсумом Эрбрана, а множество интерпретаций H-интерпретацией.

Примеры.

1. Множество дизъюнктов S = {R(x), P(g(y)) v Q(y)}, универсум Эрбрана H(S) = {a, g(a), g(g(a)), ...}, база Эрбрана A = {R(a), P(a), Q(a), R(g(a)), P(g(a)), Q(g(a)), ...}, пример H-интерпретаций:

I_1 = {R(a), P(a), R(g(a)), P(g(a)), Q(g(a)), ...},

I_2 = {¬R(a), ¬P(a), ¬Q(a), ¬R(g(a)), ¬P(g(a)), ¬Q(g(a)), ...},

I_3 = {R{a), ¬P(a), Q(a), R(g(a)), ¬P(g(a)), Q(g(a)), ...}.

2. S = {Q(x)v¬R(x), P(x), Q(b)}, H(S) = {b}, A = {Q(b), R(b), P{b)}.

I_1 = {Q(b), R(b), P(b)},

I_2 = {Q(b), R(b), ¬P(b)},

I_3 = {Q(b), ¬R(b), P(b)},

…..................................

I_8 = {¬Q(b), ¬R(b), ¬P(b)}.

Здесь множество S истинно на I_1 и I_3 и ложно на всех остальных интерпретациях.

3. S = {P(x), ¬P(x)}, H(S) = {a}, A = {P(a)}.

I_1 = {P(a)},

I_2 = {¬P(a)}.

Здесь множество S невыполнимо, так как оно ложно как в I_1, так и в I_2.

Заметим, что **эрбрановская интерпретация** (и **зрбраиовская модель**) - это интерпретация (и соответственно модель), у которой область интерпретации включает множество (предполагаемое непустым) конкретных термов рассматриваемого языка, причем любому конкретному терму сопоставляется он сам. Эрбрановская модель полностью характеризуется множеством конкретных атомов, истинных в рассматриваемой модели. Следовательно, эрбрановскую модель можно отождествлять с упомянутым множеством и сравнивать модели между собой, пользуясь теоретико-множественным отношением включения.

Семантическое дерево - это дерево, в котором каждый путь от корня к листу соответствует отдельной Н-интерпретации. Пусть, например, S = {Q(x)vR(y), P(x)}, тогда A={Q(a), R(a), P(a)}. Семантическое дерево для данной базы Эрбрана представлено на рис.2.1. Например, в листе 7 заканчивается путь, соответствующий Н-интерпретации I_7 = {¬P(a), ¬Q(a), R(a)}. Кстати, данная интерпретация не удовлетворяет S. так как атом ¬P(a) не удовлетворяет состоящему из одного предиката дизъюнкту P(x).

Вершины, в которых впервые была установлена неудовлетворительность интерпретации, называются **неблагоприятными**. Семантическое дерево, опирающееся только на неблагоприятные вершины, называется **замкнутым**. Вершина замкнутого дерева, соединенная ребрами только с неблагоприятными вершинами, называется **вершиной вывода**.

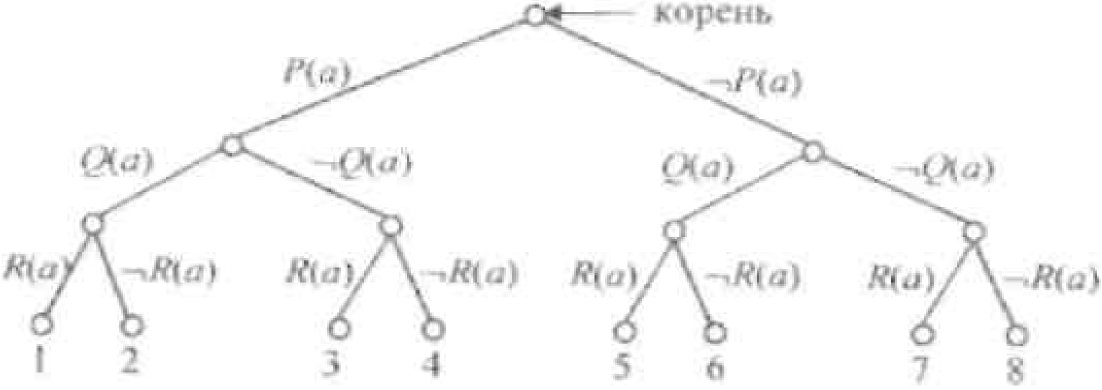


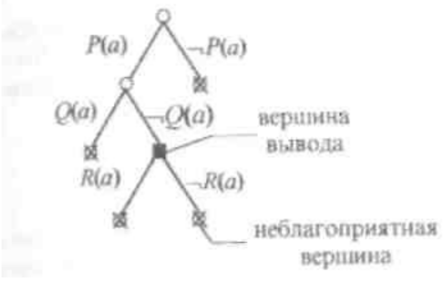
Рис. 2.1. *Семантическое дерево*

Примеры.

1. Пусть S = {P(x), Q(a)vR(y), ¬P(x)v¬Q(x), P(x)v¬R(a)}, тогда H(S) = {a},

A = {P(a), Q(a), R(a)}.

Рассмотрим все возможные интерпретации I_1, …, I_8, можно убедиться, что ни одна не удовлетворяется (рис. 2.2).



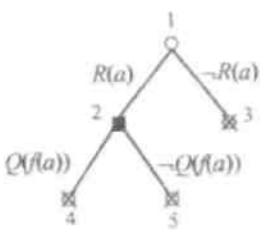


Рис. 2.2. *Замкнутое дерево* Рис. 2.3. *Замкнутое дерево*

2. Пусть S = {R(x), ¬R(x)vQ(f(x)), ¬Q(f(x)), тогда A = {R(a), Q(f(a)), Q(a), R(f(a)), Q(f(f(a))), R(f(f(a))), ...}. В вершине 3 невыполним дизъюнкт R(x), а в вершине 5 дизъюнкт ¬R(x)vQ(f(x)), в вершице 4 — дизъюнкт ¬Q(f(x)) (рис. 2.3.).

Теорема Эрбрана. *Множество дизъюнктов S невыполнимо тогда и только тогда, когда любому полному семантическому дереву для S соответствует конечное замкнутое дерево.*

Доказательство. Допустим, что невыполнимо, тогда множество атомов, приписанных ребрам соответствующей ветви, опровергает фундаментальный дизъюнкт C из S. Так как любой фундаментальный пример дизъюнкта конечен, то на ветви конечной длины существует неблагоприятная вершина. Поскольку каждая ветвь имеет неблагоприятную вершину, то для S существует замкнутое дерево. Но из каждой вершины выходит конечное число ребер, следовательно, замкнутое дерево конечно.

Наоборот, если для каждого семантического дерева для S существует конечное замкнутое дерево, то каждая его ветвь содержит неблагоприятную вершину, т.е. каждая интерпретация опровергает S. Следовательно S невыполнимо.

В литературе чаще встречается другая формулировка этой теоремы, которую приведем без доказательства.

Теорема Эрбрана (*второй вариант*). *Множество дизъюнктов невыполнимо тогда и только тогда, когда существует конечное невыполнимое множество S' фундаментальных конкретизацией дизъюнктов S.*

Важность теоремы Эрбрана состоит в том, что она гарантирует доказательство непротиворечивости за конечное число шагов. И наоборот, существующее противоречие может быть всегда достигнуто за конечное число шагов, каковы бы ни были значения истинности, даваемые функциям, присутствующим в гипотезах и заключениях.

2.5.1. Применение теоремы Эрбрана. Метод Девиса и Патнема.

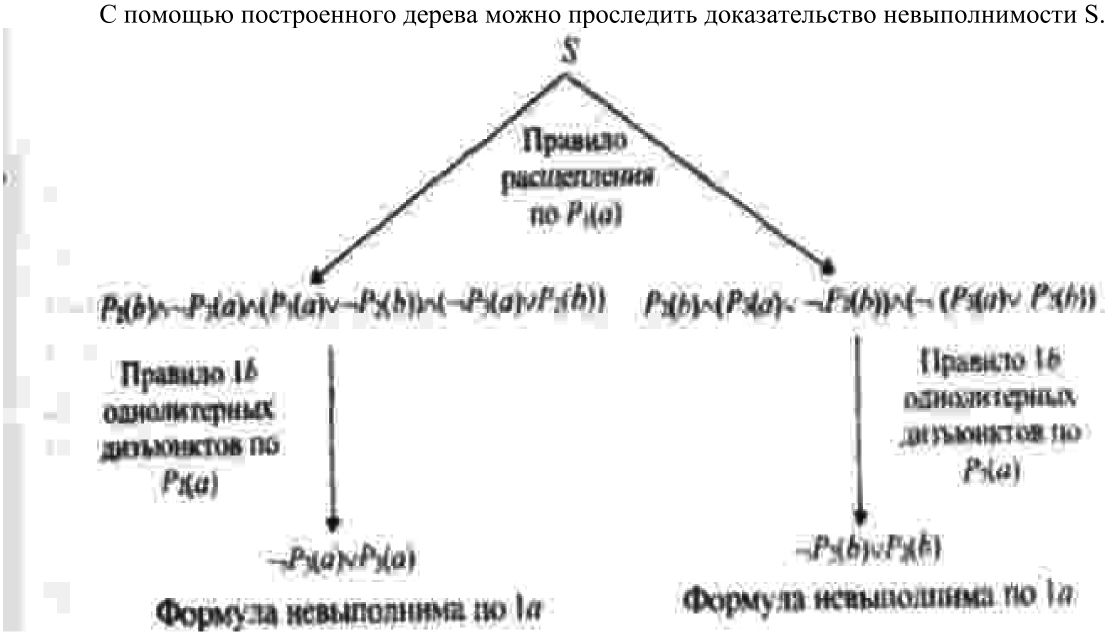
Метод Девиса-Патнема основан на втором варианте теоремы Эрбрана. Для решения вопроса о невыполнимости конечной конъюнктивной нормальной формы $S = G_1 \wedge G_2 \wedge \ldots \wedge G_n$ необходимо применить процедуру, состоящую из описанных ниже правил. На каждом шаге конъюнкция S преобразуется в новую конъюнкцию S' такую, что S невыполнима, если и только если невыполнима S'. Или порождается пара конъюнкций S' и S'' таких, что конъюнкция S невыполнима, если и только если невыполнимы обе конъюнкции S' и S''. Процедура должна завершаться за конечное число шагов, так как число различных литералов, входящих в S, на каждом шаге сокращается.

Метод составляют следующие правила:

1. *Правило однолитерных дизъюнктов.*
   a) Если в S имеется пара однолитерных основных дизъюнкта $G_j = l$ $G_j = \neg l$, то S невыполнима.
   b) Если S содержит однолитерный дизъюнкт $G_j = l$, то S' получается удалением l из S всех дизъюнктов, содержащих (включающих G_j) и затем удалением всех вхождений $\neg l$ из оставшихся основных дизъюнктов. Если S' пусто, то S выполнима.
2. *Правило чистых литер.*
   Литера l в основном дизъюнкте из S называется чистой, если $\neg l$ не встречается ни в одном основном дизъюнкте. Если литера l чистая, то удаляются все основные дизъюнкты, содержащие l. Оставшаяся часть S' невыполнима, если невыполнима S. Если не остается дизъюнктов, то S выполнима.
3. *Правило расщепления.*
   Если S можно представить в виде
   $(A_1 v l) \wedge \ldots \wedge (A_m v l) \wedge (B_1 v \neg l) \wedge \ldots \wedge (B_n v \neg l) \wedge R$, где A_i, B_i и R свободна от l и $\neg l$, то получим конъюнкции $S' = A_1 \wedge \ldots \wedge A_m \wedge R$ и $S'' = B_1 \wedge \ldots \wedge B_n \wedge R$. S невыполнима, если и только если (S'vS'') невыполнима, т.е. S' и S'' невыполнимы.
4. *Правило тавтологии.*
   Удаляются все тавтологичные основные дизъюнкты из S. Оставшаяся часть S' невыполнима тогда и только тогда, когда невыполнима S.

Пример. Пусть $S = [P_1(a) v P_2(b)] \wedge [P_1(a) v \neg P_3(a)] \wedge [\neg P_1(a) v P_3(a)] \wedge [\neg P_1(a) v \neg P_2(b)] \wedge [P_3(a) v \neg P_2(b)] \wedge [\neg P_3(a) v P_2(b)]$

С помощью построенного дерева можно проследить доказательство невыполнимости S.



Следовательно, S невыполнима.

Пример.

Пусть $S = [P_2(b)] \wedge \neg P_2(b) \wedge [\neg P_1(a) \vee \neg P_2(b) \vee \neg P_3(a)]$.

Применив правило по $\neg P_2(b)$, получим

$[\neg P_1(a) \vee \neg P_3(a)]$.

Повторив правило 1b по $P_1(a)$, получим

$\neg P_3(a)$.

Наконец, применение еще раз правила 1b по $P_3(a)$, приводит к удалению всех дизъюнктов из S. Следовательно, S выполнимо.

2.6. Метод резолюции для логики предикатов первого порядка.

По существу метод (принцип) резолюции был обоснован Ж.Эрбраном в 1930г. Первым реализовал на ЭВМ идеи Эрбрана Дж.Робинсон в 1963г.

Метод резолюции - один из методов доказательства от противного. Основная идея метода заключается в попытке вывести из множества дизъюнктов S пустой дизъюнкт, для которого нет интерпретаций и который тем самым гарантирует невыполнимость множества дизъюнктов. Процесс вывода пустого дизъюнкта может быть представлен процессом сведения числа вершин замкнутого дерева к единственной неблагоприятной вершине.

Для иллюстрации идеи вернемся, например, к замкнутому дереву на рис. 2.3. Здесь вершина вывода 2 не является неблагоприятной. Однако, если проанализировать вершины 4 и 5, в которых невыполнимы дизъюнкты ¬Q(f(x)) и ¬R(x)vQ(f(x)), соответственно, то можно сделать вывод, что одновременная выполнимость этих двух дизъюнктов возможна только в случае, если выполнен (истинен в данной модели) новый дизъюнкт. Но при добавлении в множество S дизъюнкта ¬R(x) вершина 2 становится неблагоприятной. Рассматривая теперь неблагоприятные вершины 2 и 3, в которых невыполнимы дизъюнкты ¬R(x) и R(x) соответственно, можно, рассуждая аналогично, сделать вывод, что удовлетворить одновременно этим вершинам может только пустой дизъюнкт. Для этого пустого дизъюнкта неблагоприятной вершиной является вершина 1.

Графически процесс вывода можно представить (рис.2.4) в виде *дерева вывода*, в котором листья соответствуют исходному множеству дизъюнктов S, а прочие вершины соответствуют выводимым дизъюнктам (*резольвентам*).

Другими словами, если в дизъюнктах C_1 и C_2 есть атомы P и ¬P соответственно, то они вычеркиваются, а остатки дизъюнктов объединяются и получается резольвента. Обратим внимание на то, что при переходе от семантических деревьев к деревьям вывода мы переходим от рассмотрения фундаментальных реализаций дизъюнктов к дизъюнктам, у которых термы могут содержать переменные, т.е. к наиболее общему случаю.

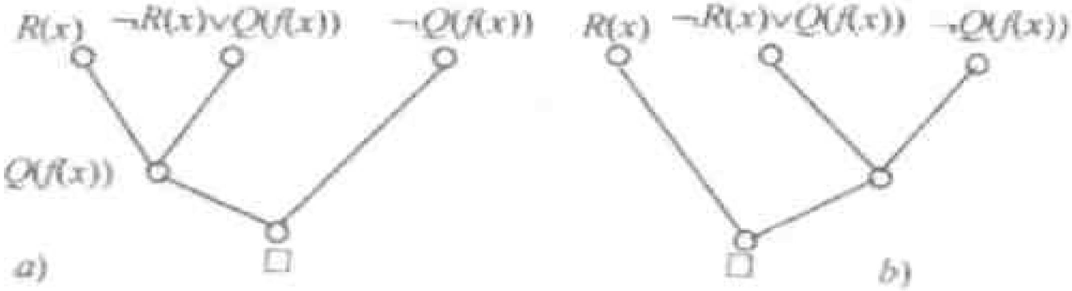


Рис. 2.4. *Дерево вывода*

Интерес представляет свойство заверши мости метода резолюции: конечное множество S невыполнимо тогда и только тогда, когда пустой дизъюнкт может быть выведен из S с помощью резолюций. Из леммы 1 (§1.6) следует достаточность этого условия. Необходимость условия отражается в следующей лемме.

Лемма 2. *Если множество дизъюнктов S невыполнимо и содержит резольвенты своих элементов, то оно обязательно содержит пустой дизъюнкт.*

Иными словами, если множество S не пусто и не содержит ни одной пары дизъюнктов, допускающих резольвенту, то оно выполнимо.

При рассмотрении семантических деревьев мы пользовались тем, что пара предикатов ¬R(x) и R(x) противоречива. Переменная x может быть заменена на константу a и противоречие высказываний ¬R(a) и R(a) становится очевидным. Вопрос, являются ли противоречивыми, скажем, предикаты ¬P(x, f(x)) и P(b, y) более сложен. На такой вопрос дает ответ процедура *унификации*.

Подстановкой будем называть конечное множество пар

$\pi = \{t_1 \mid v_1, t_2 \mid v_2, \ldots, t_n \mid v_n\}$,

где v_i - индивидные переменные, все v_i различны, t_i - термы, отличные от v_i (t_i заменяет переменную v_i).

Литерал $L'=L\pi$, полученный заменой в литерале L переменных v_i на термы t_i, L' назовется **конкретизацией** L.

Например, пусть $\pi = \{a \mid x, f(z) \mid y, b \mid z\}$, то $P(x, y, z)\pi = P(a, f(z), b)$.

Обозначим $\pi_3 = \pi_2 o \pi_1$ *композицию подстановок*, результат применения которой совпадает с результатом последовательно примененных подстановок π_2 и π_1

Определим подстановку $\pi_3 = \pi_2 o \pi_1$ через подстановки $\pi_1=\{t_1 \mid x_1, t_2 \mid x_2, \ldots, t_n \mid x_n\}$ и $\pi_2 =\{s_1 \mid y_1, s_2 \mid y_2, \ldots, s_m \mid y_m\}$.

Нетрудно заметить, что π_3 можно получить из множества $\{t_1 \pi_2 \mid x_1, t_2 \pi_2 \mid x_2, \ldots, t_n \pi_2 \mid x_n ; s_1 \mid y_1, s_2 \mid y_2, \ldots, s_m \mid y_m\}$; исключив пары $t_i \pi_2 \mid x_i$, у которых $t_i \pi_2 = x_i$, и такие пары $s_i \mid y_i$, у которых $y_i \in \{x_1, \ldots, x_n\}$.

Например, пусть $\pi_1=\{z \mid x, w \mid y, f(x) \mid z\}$ и $\pi_2 =\{a \mid x, f(y) \mid z, y \mid w\}$. В этих обозначениях имеем $x_1=x$, $x_2=y$, $x_3=z$, $y_1=x$, $y_2=z$, $y_3=w$. Тогда $\{t_1 \pi_2 \mid x_1, t_2 \pi_2 \mid x_2, t_3 \pi_2 \mid x_3, s_1 \mid y_1, s_2 \mid y_2, s_3 \mid y_3\} = \{f(y) \mid x, y \mid y, f(a) \mid z; a \mid x, f(y) \mid z, y \mid w\}$. Поскольку $t_2\pi_2=x_2$, $y_1=x_1$, $y_2=x_3$, исключаем пары $y \mid y$, $a \mid x$, $f(y) \mid z$, получим $\pi_3 = \pi_2 o \pi_1 = \{f(y) \mid x, f(a) \mid z, y \mid w\}$.

Подстановку π называют **унификатором множества** $\{L_i\}$, (i=1..n), если выполнение π приводит к равенству элементов множества $L_1\pi=L_2\pi=...=L_i\pi=...=L_n\pi$ (i=1..n). Множество $\{L_i\}$, (i=1..n) **унифицируемо**, если для него существует унификатор. Когда один из термов есть переменная или содержит переменную, но не сводится к ней, то унификация невозможна.

Пусть даны литералы $L_1=P(a, x, f(g(y)))$, $L_2=P(z, f(z), f(u))$ и подстановка $\pi=\{a \mid z, f(a) \mid x, g(y) \mid u\}$. Для подстановки π выполняется равенство

$P(a, x, f(g(y)))\pi=P(z, f(z), P(u))\pi=P(a, f(a), f(g(y)))$.

Следовательно, π - унификатор множества $\{L_1, L_2\}$.

Например, пара дизъюнкций $P(a, x)vP(b, y)$ и $P(a, f(x))vP(x, f(b))$ неунифицируемо.

Говорят, что унификатор π является более общим, чем унификатор σ, если существует такая подстановка τ, что $\sigma=\pi o \tau$.

Унификатор π называется **наиболее общим унификатором** (НОУ) тогда и только тогда, когда для каждого унификатора σ существует подстановка τ такая, что $\sigma=\tau o \pi$.

Если π - НОУ для дизъюнктивной формулы $A_1vA_2v...vA_n$, то атомарная формула $B= A_1\pi=A_2\pi=...=A_n\pi$ называется **фактором** для $A_1vA_2v...vA_n$.

2.6.1. Алгоритм унификации.

Для нахождения НОУ используется *алгоритм унификации*, который за конечное число шагов определяет НОУ, если он существует.

Два выражения P(x) и P(a) не тождественны. Они различаются аргументами. Это различие назовем рассогласованием и обозначим {x, a}. Дадим следующее определение: **множеством рассогласований** дизъюнктивной атомарной формулы $P(v_1^{(1)}, \dots, v_n^{(1)}) \vee P(v_1^{(2)}, \dots, v_n^{(2)}) \vee \dots \vee P(v_1^{(N)}, \dots, v_n^{(N)})$ или короче - $A_1 \vee A_2 \vee \dots \vee A_N$ есть множество $\{t(1), t^{(2)}, \dots, t^{(N)}\}$, которое получается следующим образом.

Просматриваются все атомарные формулы A_i слева и выявляется первая позиция, на которой не для всех A_i стоит один и тот же символ. Тогда для каждого i ($1 \leq i \leq N$) рассогласование $\tau^{(i)}$, есть различающиеся подвыражения из всех Ah начиная с этой позиции. Например, для дизъюнкции

$P(x, g(\underline{f(y, z)}, x), y) \vee P(x, g(\underline{a}, b), b) \vee P(x, g(\underline{g(h(x), a)}, y), h(x))$ находим, просматривая слева цепочку символов, седьмую позицию, с которой символы различаются (подчеркнуто) и $t^{(2)} = \{f(y, z), a, g(h(x), a)\}$. Заметим, что множество рассогласований может содержать менее N элементов, так как не все $t^{(i)}$ должны быть отличающимися.

Далее находится и выполняется подстановка $\pi_1 = (t \mid x)$, если множество рассогласований содержит переменную x и терм t, который не содержит x. В противном случае рассматриваемое множество литер не унифицируемо. Аналогично находится и выполняется следующая подстановка: π_2, а результирующая подстановка равна композиции подстановок, найденных на каждом шаге работы алгоритма.

Основная часть метода резолюции заключается в использовании алгоритма унификации, который находит наиболее общий унификатор π для унифицируемой дизъюнктивной формулы и сообщении о завершении, если формула не унифицируема. Последовательность шагов алгоритма унификации следующая. Пусть задана дизъюнктивная формула A или задано множество дизъюнктов, подлежащих унификации, т.е. получения фактора для формулы A.

Шаг 1. $A_K = A$, k=0, подстановка $\alpha_K = \alpha_0$.

Шаг 2. Если A_K - фактор (единичный дизъюнкт), то α_K - НОУ и процедура завершается. Иначе формируем множество рассогласований D_K для A_K.

Шаг 3. Если существуют такие v_K и t_K в D_K, что v_K - индивидная переменная, не входящая в t_K, то перейти к шагу 4. В противном случае - формула (множество) A не унифицируема, процедура завершается.

Шаг 4. $\alpha_{K+1} = \alpha_K\{t_K \mid v_K\}$ и $A_{K+1} = A_K\{t_K \mid v_K\}$, k=k+1. Перейти к шагу 2.

Пример. Найти НОУ для множества

$L = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}$.

На первом шаге имеем множество рассогласований $D_1 = \{a, z\}$ и подстановку $\alpha_1 = (a \mid z)$.

В результате выполнения подстановки получаем множество литер

$L_1 = L\alpha_1 = \{P(a, x, f(g(y))), P(a, f(a), f(u))\}$.

На втором шаге имеем $D_2 = \{x, f(a)\}$, $\alpha_2 = \{f(a) \mid x\}$ и

$L_2 = L_1\alpha_2 = \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}$.

На третьем и последнем шаге имеем $D_3 = \{g(y), u\}$, $a_3 = \{g(y) \mid u\}$

$L_3 = L_2\alpha_3 = \{P(a, f(a), f(g(y)))\}$.

В результате найдем НОУ множества

$\pi = \alpha_3 o \alpha_2 o \alpha_1 = \{a \mid z, f(a) \mid x, g(y) \mid u\}$.

Пример. Найти НОУ для дизъюнкции

$P(f(a), g(x)) \vee P(y, y)$ или короче - $A_1 \vee A_2$.

Начнем с $\alpha_0 = \{\}$ - пустая подстановка. Множество рассогласований для $A = A_1 \vee A_2$ есть $D_0 = \{f(a), y\}$. Из этого следует $v_0 = y$, $t_0 = f(a)$. Формируем подстановку $\alpha_1 = \{f(a) \mid y\}$.

После подстановки дизъюнкция принимает вид:

$A\alpha_1 = A_1\alpha_1 \vee A_2\alpha_1 = P(f(a), g(x)) \vee P(f(a), f(a))$.

Рассогласование для $A\alpha_1$ есть $D_1 = \{g(x), f(a)\}$.

В D_1 нет элемента, который мог бы быть переменной. Поэтому алгоритм унификации завершается; делаем заключение -исходный дизъюнкт A не унифицируем.

Унификация используется для решения двух задач :

1. Унифицирует одноименные предикаты из разных дизъюнктов, отличающиеся наличием отрицания. В результате становится возможным получение резольвенты.
2. Унифицирует одноименные предикаты из одного дизъюнкта, имеющие одинаковый знак. В результате становится возможным получение частного случая для данного дизъюнкта (фактора). **Факторы** - это резольвенты, полученные на базе одного дизъюнкта.

Так, из дизъюнкта R(x, y)vR(a, f(x))vR(x, x) можно получить факторы
R(a, f(a))vR(a, a)　　　　R(x, x)vR(a, f(x)).

Приведем следующие важные для дальнейшего понимания правил резолюции определения.

Определение 1. Если две или более литер (с одинаковы знаком) дизъюнкта A имеют наиболее общий унификатор НОУ σ, то Aσ называется **склейкой** A. Если Aσ - единичный дизъюнкт, то склейка называется **единичной**.

Пример. Пусть A = $\underline{P(x)}$v$\underline{P(f(y))}$v¬Q(x). Тогда подчеркнутые литеры имеют НОУ σ = {f(y) | x}. Следовательно, Aσ = $\underline{P(f(y))}$v¬Q($\underline{f(y)}$) - склейка A.

Определение 2. Пусть A_1 и A_2 - два дизъюнкта (посылки), которые не имеют общих переменных. Пусть C_1 и C_2 две литеры из A_1 и A_2 соответственно. Если C_1 и C_2 имеют НОУ σ, то дизъюнкт (A_1σ-C_1σ)v(A_2σ-C_2σ) называется бинарной резольвентой A_1 и A_2. Литеры C_1 и C_2 - отрезаемые литеры.

Пример.

A_1 = P(x)vQ(x), A_2 = ¬P(a)vR(x).

Так как x входит в A_1 и A_2, то заменим x в A_2 и пусть A_2 = ¬P(a)vR(y). Выберем C_1 = P(x) и C_2 = ¬P(a). НОУ для этих литер есть σ = {a | x}. Следовательно,
(A_1σ-C_1σ)v(A_2σ-C_2σ) = ({P(a), Q(a)}-{P(a)}v{¬P(a), R(y)}-{¬P(a)}) = {Q(a)vR(y)} = Q(a)vR(y).

Дизъюнкт Q(a)vR(y) ~ бинарная резольвента, а P(x) и ¬P(a) - обрезаемые литеры.

2.6.2. Правило резолюций.

Метод резолюции для доказательства выводимости теоремы из множества S дизъюнктов основан на применении (возможно неоднократном) правила резолюции, состоящего из последовательных действий, приведенных ниже. Это правило, представляющее содержание метода резолюции, является отражением леммы, сформулированной в начале раздела. По существу правило резолюций есть последовательность шагов логического вывода, порождающего резольвенты для множества
дизъюнктов.

Лучшей иллюстрацией применения этого правила являются примеры, в которых подробно рассматривается последовательность действий.

Сформулируем основные положения:

1. Модель исследуемого "мира" представляется множеством аксиом, которые преобразуются в множество дизъюнктов S.
2. Для доказательства справедливости теоремы (B) в дан ном "мире" необходимо взять ее отрицание и, преобразовав в форму дизъюнкта (или дизъюнктов), добавить к множеству S. Если теорема верна (выводима), то новое множество дизъюнктов (вместе с отрицанием теоремы) должно быть противоречиво.
3. Доказательство противоречивости сводится к доказательству того, что из данного множества дизъюнктов может быть выведен пустой дизъюнкт.
4. Чисто технически метод резолюции состоит из унификаций и получения множества резольвент до тех пор, пока не будет получена пустая резольвента.
5. Для уменьшения числа резольвент (а, следовательно, для повышения эффективности вывода) очень важна стратегия вывода, т.е. определение того, в какой последовательности выбирать дизъюнкты для получения резольвент.
6. Если множество дизъюнктов S противоречиво, то пустой дизъюнкт будет найден за конечное число шагов. Однако, если множество S непротиворечиво, процесс установления этого факта может быть бесконечным.

Пример. Показать выполнимость формулы А.

А: $\neg\exists y\forall z[P(z, y)\leftrightarrow\neg\exists x[P(z, x)\wedge P(x, z)]]$.

Методом резолюции докажем невыполнимость формулы $\neg$A. Для этого сперва исходную формулу приведем к дизъюнктивной форме.

a) Исключим логическую связку $\leftrightarrow$:
$\exists y\forall z\{[P(z, y)\rightarrow\neg\exists x[P(z, x)\wedge P(x, z)]]\wedge[\neg\exists x[P(z, x)\wedge P(x, z)] \rightarrow P(z, y)]\}$.
b) Исключим логическую связку $\rightarrow$:
$\exists y\forall z\{[\neg P(z, y)\vee\neg\exists x[P(z, x)\wedge P(x, z)]]\wedge[\exists x[P(z, x)\wedge P(x, z)]\vee P(z, y)]\}$.
c) Внесем внешнее отрицание в скобки:
$\exists y\forall z\{[\neg P(z, y)\vee\forall x[\neg P(z, x)\vee\neg P(x, z)]]\wedge[\exists x[P(z, x)\wedge P(x, z)]\vee P(z, y)]\}$.
d) Вынесем за скобки кванторы $\exists$ и $\forall$, сделав соответствующую замену переменных:
$\exists y\forall z\exists v\forall x\{[\neg P(z, y)\vee[\neg P(z, x)\vee\neg P(x, z)]]\wedge[[P(z, v)\wedge P(x, z)]\vee P(z, y)]\}$.
e) Удалим кванторы существования (сколемизация):
$\{[\neg P(z, a)\vee\neg P(z, x)\vee\neg P(x, z)] \wedge [P(z, f(z))\wedge P(f(z), z)]\vee P(z, a)]\}$.
f) Применим к правой скобке дистрибутивный закон:
$\{[\neg P(z, a)\vee\neg P(z, x)\vee\neg P(x, z)] \wedge [P(z, f(z))\vee P(z, a)]\}$.

Выпишем полученные дизъюнкты, изменив в каждом из них имена переменных:

1. $\neg P(z_1, a) \vee \neg P(z_1, x_1) \vee \neg P(x_1, z_1)$
2. $P(z_2, f(z_2)) \vee P(z_2, a)$
3. $P(f(z_3), z_3) \vee P(z_3, a)$

   Попробуем найти резольвенту для дизъюнктов 1 и 2. Претендентом на резольвирование является литерал $P(z_2, a)$ из дизъюнкта 2. Поэтому для этого построим дизъюнктивную формулу $\neg P(z_1, a) \vee \neg P(z_1, x_1) \vee \neg P(x_1, z_1) \vee P(z_2, a)$. Для этой формулы наиболее общим унификатором является

   $\alpha_1 = \{a \mid x_1, a \mid z_2, a \mid z_1\}$.

   После подстановки в 1 и 2 получим резольвенту
4. $P(a, f(a))$.

   Для дизъюнктов 1 и 3 НОУ есть $\alpha_2 = \{a \mid x_1, a \mid z_1, a \mid z_3\}$. Резольвентой для 1 и 3 является
5. $P(f(a), a)$.

   Для 1 и 4 НОУ есть $\alpha_3 = \{a \mid x_1, f(a) \mid z_1\}$. Получим резольвенту
6. $\neg P(f(a), a)$.

   Из 5 и 6 имеем
7. $\square$.

Теперь можно сделать заключение: ¬A не выводима, следовательно исходная A формула значима.

Пример. Показать значимость формулы:

A: $\exists x \exists y \forall z\{[P(x, y) \rightarrow [P(y, z) \wedge P(z, z)]] \wedge [[P(x, y) \wedge Q(x, y)] \rightarrow [Q(x, z) \wedge Q(z, z)]]\}$

Докажем невыводимость ¬A. Как и в предыдущем примере, приведем формулу ¬A к дизъюнктивной форме.

a) a) Исключим логическую связку →:

   $\neg \exists x \exists y \forall z\{[\neg P(x, y) \vee [P(y, z) \wedge P(z, z)]] \wedge [\neg [P(x, y) \wedge Q(x, y)] \vee [Q(x, z) \wedge Q(z, z)]]\}$

b) Внесем внешнее отрицание в скобки:

   $\forall x \forall y \exists z\{[P\{x, y) \wedge [\neg P(y, z) \vee \neg P(z, z)]] \vee [[P(x, y) \wedge Q(x, y)] \wedge [\neg Q(x, z) \vee \neg Q(z, z)]]\}$

c) Удалим кванторы существования (сколемизация):

   $\{[P(x, y) \wedge [\neg P(y, f(x, y)) \vee \neg P(f(x, y), f(x, y))]] \vee$
   $[(P(x, y) \wedge Q(x, y)) \wedge [\neg Q(x, f(x, y)) \vee \neg Q(f(x, y), f(x, y))]]\}$

d) Применим дистрибутивный закон ко всему выражению.

   Преобразование выполняется в соответствии со схемой:

[insert image here]

$[P(x, y) \vee P(x, y)] \wedge [P(x, y) \vee Q(x, y)] \wedge [P(x, y) \vee \neg Q(x, f(x, y)) \wedge \neg Q(f(x, y), f(x, y))] \wedge$
$[\neg P(y, f(x, y)) \vee \neg P(f(x, y), f(x, y)) \vee P(x, y)] \wedge$
$[\neg P(y, f(x, y)) \vee \neg P(f(x, y), f(x, y)) \vee Q(x, y)] \wedge [\neg P(y, f(x, y)) \vee \neg P(f(x, y), f(x, y)) \vee$
$\neg Q(x, f(x, y)) \vee \neg Q(f(x, y), f(x, y))]$

e) Упрощение: из первого дизъюнкта удаляется одно вхождение $P(x, y)$, удаляются все дизъюнкты, содержащие $P(x, y)$, т.е. 2-й, 3-й и 4-й. Выпишем полученные дизъюнкты, заменив имена переменных в каждом дизъюнкте:

   1. $P(x_1, y_1)$.
   2. $\neg P(y_2, f(x_2, y_2)) \vee \neg P(f(x_2, y_2), f(x_2, y_2)) \vee Q(x_2, y_2)$
   3. $\neg P(y_3, f(x_3, y_3)) \vee \neg P(f(x_3, y_3), f(x_3, y_3)) \vee \neg Q(x_3, f(x_3, y_3)) \vee \neg Q(f(x_3, y_3), f(x_3, y_3))$.

      Для 1 и 2 находим унификатор $\{y_2 \mid x_1, f(x_2, y_3) \mid y_1\}$ и (после подстановки) резольвенту:
   4. $\neg P(f(x_2, y_2), f(x_2, y_2)) \vee Q(x_2, y_2)$

      Резольвируют дизъюнкты 1 и 4: унификатор $\{f(x_2, y_2) \mid x_1, f(x_2, y_2) \mid y_1\}$ и резольвента
   5. $Q(x_2, y_2)$

      Дизъюнкты 3 и 5 с унификатором $\{f(x_3, y_3) \mid x_2, f(x_3, y_3) \mid x_2)$ дадут резольвенту:
   6. $\neg P(y_3, f(x_3, y_3)) \vee \neg P(f(x_3, y_3), f(x_3, y_3)) \vee \neg Q(x_3, f(x_3, y_3))$

      Теперь резольвируют дизъюнкты 5 и 6 с унификатором $\{x_3 \mid x_2, f(x_3, y_3) \mid y_2\}$:
   7. $\neg P(y_3, f(x_3, y_3)) \vee \neg P(f(x_3, y_3), f(x_3, y_3))$

Для дизъюнктов 1 и 7 с унификатором $\{f(x_3, y_3) \mid x_1, f(x_3, y_3) \mid y_3\}$ получим резольвенту:

8. $\neg P(y_3, f(x_3, y_3))$

Наконец, для дизъюнктов 1 и 8 с унификатором $\{y_3 \mid x_1, f(x_3, y_3) \mid y_1\}$ получаем

9. □.

Таким образом формула ¬A не выводима, следовательно, праведливо исходное утверждение.

Рассмотрим несколько примеров применения метода резолюции к поиску ответов на вопросы. Разделим вопросы на четыре класса в зависимости от формы ответа:

Класс А. Вопросы, требующие ответа "да " или " нет".

Класс В. Вопросы, требующие в качестве ответа "где?", "кто?" или "при каких условиях?".

Класс С. Вопросы, требующие ответа в виде последовательности действий.

Класс D. Вопросы, включающие проверку условий, например, имеющие конструкцию "ЕСЛИ <условие>,' ТО <действие>".

Вопросы класса А.

Пример.

Аксиомы:

1. Для всех x, y, z, если x и y - братья, а также y и z - братья, то x и z - братья.
2. Борис и Кирилл - братья.
3. Кирилл и Мефодий - братья.
4. Мефодий и Глеб — братья.

Доказать теорему

5. Борис и Глеб — братья.

Запишем аксиомы на языке предикатов. Пусть P(x, y) означает "x и y - братья". Тогда постановку задачи можно представить в форме предикатов:

A_1: $P(x, y) \wedge P(y, z) \rightarrow P(x, z) \equiv \neg[P(x, y) \wedge P(y, z)] \vee P(x, z) \equiv \neg P(x, y) \vee \neg P(y, z) \vee P(x, z)$.

A_2: P(Б, К).

A_3: P(К, М).

A_4: P(М, Г).

B_5: P(Б, Г).

Представим аксиому в виде дизъюнктов, кроме того, теорему B_5, которую требуется доказать, необходимо взять с отрицанием.

$Д_1$: $\neg P(x, y) \vee \neg P(y, z) \vee P(x, z)$.

$Д_2$: P(Б, К).

$Д_3$: P(К, М).

$Д_4$: P(М, Г).

$Д_5$: ¬P(Б, Г).

Вывод изобразим в виде дерева вывода

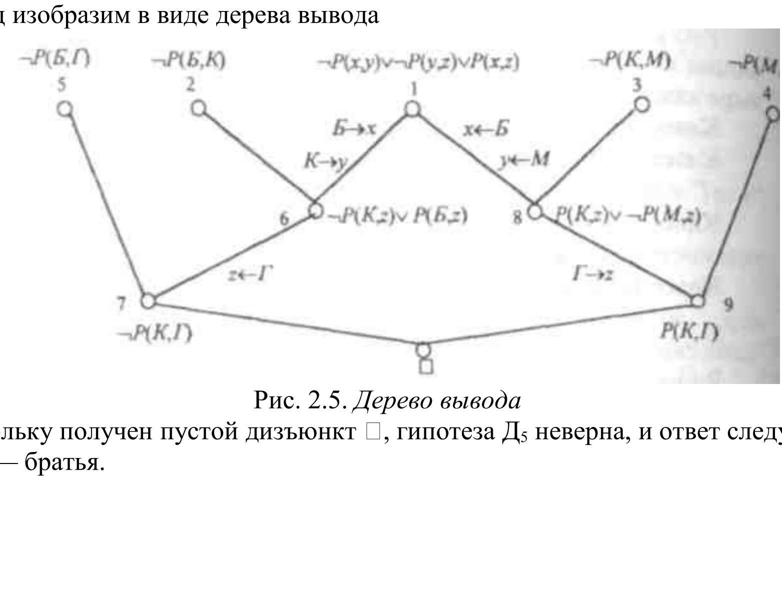


Рис. 2.5. *Дерево вывода*

Поскольку получен пустой дизъюнкт □, гипотеза $Д_5$ неверна, и ответ следует "да", т.е. Борис и Глеб — братья.

Вопросы класса В.

Пример.

Аксиомы:

1. Миша повсюду ходит за Димой.
2. Дима в школе.

Доказать теорему:

3. Где Миша?

На языке предикатов:

A_1: $\forall x(B(\textit{Дима},x)\rightarrow B(\textit{Миша}, x))$.

A_2: $B(\textit{Дима}, \textit{школа})$.

B_3: $\neg\exists x B(\textit{Миша}, x) \equiv \forall x\neg B(\textit{Миша}, x)$.

Аналогично рассмотренному в предыдущем примере имеем:

$Д_1$: $\neg B(\textit{Дима}, x)\vee B(\textit{Миша}, x)$.

$Д_2$: $B(\textit{Дима, школа})$.

$Д_3$: $\neg B(\textit{Миша}, x)$.

Для получения ответа введем специальный дизъюнкт ОТВ, термы которого дублируют термы предиката вопроса. С учётом сказанного дизъюнкт $Д_3$ следовало бы заменить на дизъюнкт $Д_3'$: $\neg B(\textit{Миша}, x)\vee ОТВ(\textit{Миша}, x)$.

В этом и в последующих примерах будем пользоваться не графическим изображением резольвент, а записью их последовательности.

($Д_1$ - $Д_2$): $Д_4$. $B(\textit{Миша}, \textit{школа})$.

($Д_3'$- $Д_4$): $Д_5$. $\square\vee ОТВ(\textit{Миша}, \textit{школа})$, т.е. предикат ОТВ наследует смысл предиката вопроса и в данном случае читается как "Миша в школе".

Пример.

Аксиомы:

1. Для всех x, y, и z, если x - отец y, z - отец x, то z - прародитель y
2. Каждый человек имеет своего родителя.

Вопрос:

3. Для каждого x, кто является прародителем x?

Формальная запись будет следующей:

A_1: $\forall x, y, z(P(x, y)\wedge P(z, x)\rightarrow П(z, y)$.

A_2: $\forall x\exists v(P(v, x))$.

B_3: $\forall x П(u, x)$.

Преобразуем аксиомы в дизъюнкты (взяв отрицание B_3):

$Д_1$. $\neg P(x, y)\vee\neg P(z, x)\vee П(z, y)$.

$Д_2$. $P(f^C(x), x)$.

$Д_3$. $\neg П(u, x)\vee ОТВ(u, x)$.

Сколемовскую функцию f^C, введенную при исключении квантора существования, можно интерпретировать как имя родителя каждого индивида, поэтому при выводе резольвент заменой переменных подчеркнем изменение этого имени.

Получение ответа :

В дизъюнкте $Д_2$ сделаем замену переменных $v\rightarrow x$

($Д_2$: $P(f^C(v), v)$. Тогда при сопоставлении дизъюнктов $Д_1$ - $Д_2$ для получения резольвенты делаем подстановку $z\leftarrow f^C(v)$, $x\leftarrow v$.

($Д_1$ - $Д_2$): $Д_4$. $\neg P(v, y)\vee П(f^C(v), y)$

Теперь при сопоставлении дизъюнктов $Д_4$ - $Д_2$ для получения новой резольвенты делаем новую замену v в $Д_2$ на q.

($Д_4$ - $Д_2$): $Д_5$. $П(f^C(f^C(q)), q)$ $\quad$ ($v\leftarrow f^C(q)$, $y\leftarrow q$).

($Д_5$ - $Д_3$): $Д_6$. $\square\vee ОТВ(f^C(f^C(q)), q)$ $\quad$ ($x\leftarrow q$, $u\leftarrow f^C(f^C(q))$)

Полученный ответ можно интерпретировать следующим образом:

«Для любого индивида (x) родитель родитель этого индивда x является его прародителем».

К сожалению, в каждом конкретном случае функция Сколема интерпретируется, исходя из смысла вопроса и используемых в модели предикатов. Формализовать этот процесс сложно (если возможно вообще). В данном случае $f^C(x)$ - "быть родителем x".

<u>Вопросы класса С</u>.

Для вопросов этого типа задача состоит в нахождении последовательности действий, достигающей некоторой цели. Важное понятие, используемое в нахождении ответов на вопросы этого типа, - это так называемый метод *состояний* и их *преобразований*. Считается, что каждый из рассматриваемых объектов находится в данный момент в определенном состоянии. Для достижения цели мы должны изменить текущее состояние объекта в желаемое. Рассмотрим возможность автоматического доказательства теорем для нахождения таких действий.

Сформулируем задачу. Пусть начальное состояние объекта d является s_1 и d вначале находится в точке a. Обозначим P(x, y, z) - “x находится в точке y в состоянии z”. Перемещение объекта в другую точку под действием /i приводит к изменению исходного состояния.

Пример.

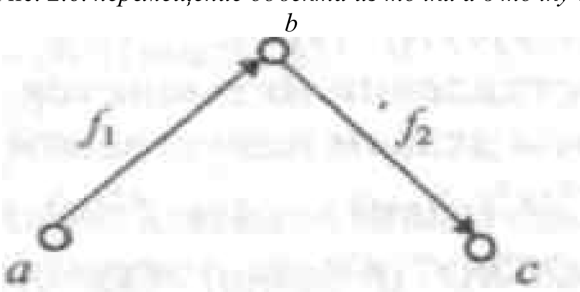
Аксиомы:

1. Объект d находится в точке a в состоянии s_1.
2. Объект перемещается из исходной точки в точку b под воздействием f_1
3. Объект перемещается из точки b в точку c под воздействием f_2.

Вопрос:

4. Как передвинуть d из a в c, если прямого пути из a в c нет (см. рис. 2.6)?

Рис. 2.6. *перемещение объекта из точки a в точку c*



Формальная запись будет следующей:

A_1: $P(d, a, s_1)$.

A_2, A_3: $(\forall x, y_i, z, \exists y_j)(P(x, y_i, z) \rightarrow P(x, y_j, f_i(x, y_i, y_j, z)))$.

B_4: $\exists x, y, zP(x, y, z)$.

Преобразуем аксиомы в дизъюнкты (взяв отрицание вопроса):

$Д_1$. $P(d, a, s_1)$.

$Д_2$. $\neg P(d, a, z) v P(d, b, f_1(d, a, b, z))$.

$Д_3$. $\neg P(d, b, z) v P(d, c, f_2(d, a, c, z))$.

$Д_4$. $\neg P(d, c, a) v OTB(z)$.

Решение:

$Д_1$ - $Д_2$: $Д_5$. $P(d, b, f_1(d, a, b, s_1))$ $(z \leftarrow s_1)$

$Д_3$ - $Д_5$: $Д_6$. $P(d, c, f_2(d, b, c, f_1(d, a, b, s_1)))$ $(z \leftarrow f_1(d, a, b, s_1))$.

$Д_6$ - $Д_4$: $Д_7$. $\square v OTB(f_2(d, b, c, f_1(d, a, b, s_1)))$ $(z \leftarrow f_2(d, b, c, f_1(d, a, b, s_1)))$

Дизъюнкт $OTB(f_2(d, b, c, f_1(d, a, b, s_1)))$ можно интерпретировать как выполнение двух действий - f_1 и f_2. Иными словами, сначала применяется действие f_1 для перемещения объекта d из a в b, а затем применяется действие f_2 для перемещения d из b в c.

<u>Вопросы класса Д</u>.

Аксиомы:

1. Если Пете меньше пяти лет, то он должен принимать лекарство a.
2. Если Пете не меньше пяти лет, то он должен принимать лекарство b.

Вопрос:

3. Какое лекарство должен принимать Петя?

Пусть P(x) - "человеку x меньше пяти лет", R(x, y) - "x должен принимать лекарство y".
Перейдем сразу к дизъюнктивной форме:
$Д_1$. ¬P(*Петя*)vR(*Петя*, a).
$Д_2$. P(*Петя*)vR(*Петя*, b).
$Д_3$. ¬R(*Петя*, x)vOTB(x).

Решение:
$Д_1$ - $Д_3$: $Д_4$. ¬P(*Петя*)vOTB(a).
$Д_2$ - $Д_3$: $Д_5$. P(*Петя*)vOTB(b).
$Д_4$ - $Д_5$: $Д_6$. OTB(a)vOTB(b).

Мы можем задать вопрос: "При каком условии будет истинен OTB(a)?" или "При каком условии OTB(a) будет логическим следствием дизъюнктов $Д_1$, $Д_2$, $Д_3$". Аналогичный вопрос можно задать для дизъюнкта OTB(b). Покажем, как можно получить информацию, анализируя вывод дизъюнкта-ответа.

Алгоритм извлечения информации.
Обозначим T_0 - дерево вывода, соответствующее полученному решению.
Шаг 1. Припишем ребрам резольвирующих узлов дерева T_0 отброшенные предикаты с соответствующей подстановкой переменных и обозначим это дерево T_1 (рис.2.7).

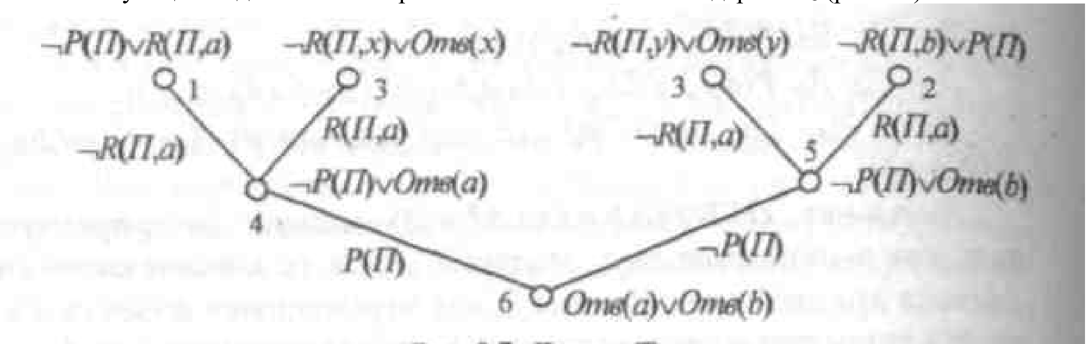


Рис. 2.7. Дерево T_1

Шаг 2. Перевернем дерево, добавим стрелки к ребрам и выбросим все дизъюнкты, приписанные узлам. Получим дерево (рис.2.8), где прописными буквами помечены листья (аргумент П — *Петя*).

Шаг 3. В дереве T_2 удалим все узлы (и связанные с ними ребра), соответствующие дизъюнктам, не содержащим предиката OTB (листья B, C и связанные с ними ребра). Результирующее дерево T_3 изображено на рис.2.9.

Шаг 4. Пусть N_1, N_2, …, N_m - листья дерева T_3. Для каждого N_i, i=1..m пусть $I(N_i)$ обозначает конъюнкцию литералов, приписанных пути (ребрам) от самого верхнего узла к N_i Пусть $C(N_i)$ - дизъюнкт, соответствующий узлу N_i. Находим в дизъюнкте-ответе литерал $L(N_i)$ такой, что $L(N_i)$ - логическое следствие конъюнкции $I(N_i)$^$C(N_i)$. Припишем $L(N_i)$ узлу N_i

В нашем примере C(A)=¬R(П, x)vOTB(x) и L(A)=OTB(a), так как I(A)=P(П)^R(П, a). Аналогично L(B)=OTB(b). Полученное таким образом дерево T_4 изображено на рис.2.10.

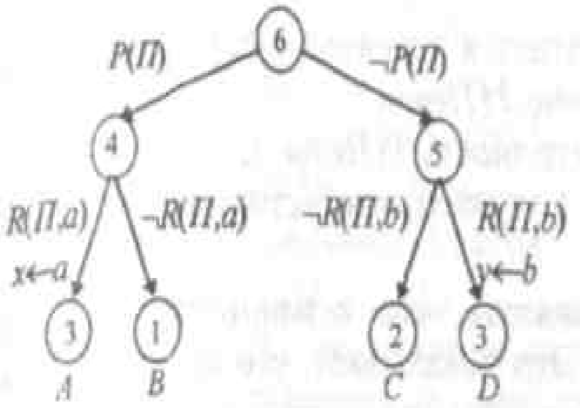


Рис. 2.8. Дерево T_2

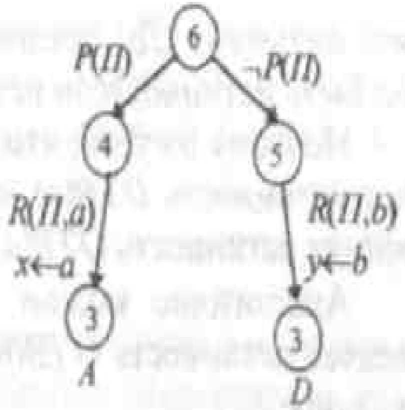


Рис. 2.9. Дерево T_3

Шаг 5. Запишем в виде $N_1, N_2, \ldots, N_q$ список всех узлов N_i дерева T_4 таких, что из $1 \le i \le q$ ведет только одно ребро r_i. Удалим из дерева литералы $L(r_i)$, где $L(r_i)$ - литерал приписанный ребру r_i Обозначим результирующее дерево T_5. Для нашего примера T_5:

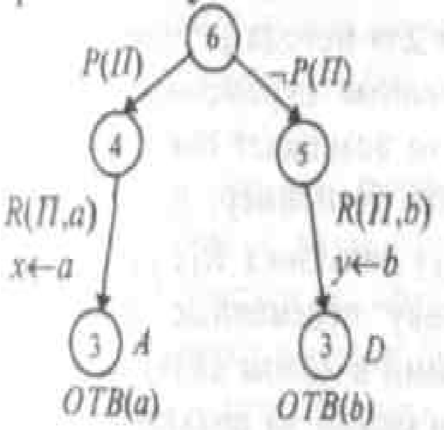


Рис. 2.10. Дерево T_4

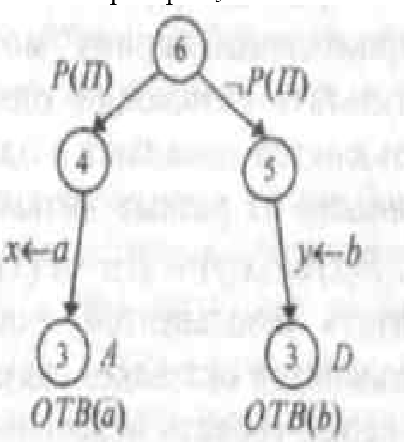


Рис. 2.11. Дерево T_5

Заметим, что рис. 2.11 удобно рассматривать как дерево решений. Если истинно P(П), то истинно ОТВ(а). В противном случае истинно ОТВ(b). Таким образом, наш ответ таков: 'Если Пете меньше пяти лет, то он должен принимать лекарство а. В противном случае - принимать лекарство b'.

На шаге 4 мы приписали ОТВ(а) узлу А после того, как было обнаружено, что ОТВ(а) - логическое следствие из P(*Петя*)^R(*Петя*, а)^(¬R(*Петя*, x)vОТВ(x)). Так как дизъюнкт (¬R(*Петя*, x)vОТВ(x)) входит в число исходных дизъюнктов, он предполагается всегда истинным. Поэтому ОТВ(а) истинен всегда, когда истинна конъюнкция P(*Петя*)^R(*Петя*, а). Рассмотрим дерево T_2. Узел В соответствует дизъюнкту ($Д_1$). Так как он не содержит предиката ОТВ, все входящие в него литеры будут отброшены при получении дизъюнкта-ответа. Поскольку отрицания всех этих дизъюнктов содержится в I(B), где I(B)={P(*Петя*), R(*Петя*, а)}, нетрудно показать, что R(*Петя*, а) является логическим следствием P(*Петя*) и дизъюнкта ($Д_1$). Так как дизъюнкт ($Д_1$) предполагается истинным, R(*Петя*, а) должно быть истинно, если истинно P(*Петя*).

Поэтому из того, что истинность P(*Петя*) и R(*Петя*, а) влечет истинность ОТВ(а) мы получаем, что истинность P(*Петя*) влечет истинность ОТВ(а).

Аналогично можно показать, что истинность ¬P(*Петя*) влечет истинность ОТВ(b). Это доказывает, что дерево T_5 правильное.

В заключение отметим, что для некоторого упрощения резолюции в ручном варианте мы не выполняли обязательную при машинной реализации процедуру переименования переменных. Переименование переменных обусловлено тем, что одноименные переменные могут и должны быть связаны только в пределах одного дизъюнкта. Каждый дизъюнкт - это "строительный кирпич" модели для метода резолюции. Если в результате резолюций одноименные переменные из разных дизъюнктов попадают в один, то возникает связь между переменными из разных дизъюнктов. Например, для дизъюнктов $Д_1$: P(x)vR(x, y) и $Д_2$: ¬P(x)vP(y) дизъюнкт R(x, y)vP(y) нельзя считать резольвентой, поскольку переменные y из разных дизъюнктов оказалась связанными в новом дизъюнкте. Следует переименовать переменные в одном из дизъюнктов, например в $Д_1$, тогда получим $Д_1$': P(z)vR(z,v).

В результате унификации {z | x} (или {x | z}) и последующей резолюции получим резольвенту $Д_3$': R(z, v)vP(z). Разумеется, переменная v должна быть определена на том же множестве, что и переменная y.

2.7. Формы представления логических формул.

2.7.1. Клаузальные формы.

В целях формализации доказательства выводимости в предыдущих разделах были рассмотрены преобразования правильно построенных формул в *предваренную нормальную форму* (кванторы находятся в передней части формулы). Предваренная нормальная форма преобразовывалась в *нормальную сколемовскую форму* исключением кванторов существования и введением специальным образом сколемовских функций и констант.

В искусственном интеллекте часто бывает полезно представлять формулы в *клаузальной форме*. Сколемовская нормальная форма может быть преобразована в клаузальную форму, т. е. в виде множества *дизъюнктов*. Это позволяет иногда упростить доказательство теорем. **Дизъюнкт** - это дизъюнкция литер, в которых переменные неявным образом универсально квантифицировны.

Общий вид дизъюнкта таков:

$\neg P_1 v ... v \neg P_m v Q_1 v ... v Q_n$.

Здесь P_i и Q_j - позитивные литералы (атомарные формулы, в которых все переменные предполагаются универсально квантифицированными). Дизъюнкт без переменных называется конкретизированным дизъюнктом. Приведенный выше дизъюнкт можно записать в эквивалентной форме:

$P_1 \wedge ... \wedge P_m \rightarrow Q_1 v ... v Q_n$.

В зависимости от числа литер в антецеденте (посылке) и в консеквенте (заключении) этой импликации имеем различные по выразительности подмножества языка: логический эквивалент языка реляционных баз данных, логический язык, используемый в Прологе и др.

Когда n позитивных литералов в дизъюнкте строго больше 1, то имеем дело с дизъюнктивной информацией (возможно, условной). В частности, при $m=0$ и $n>1$ дизъюнкт содержит только позитивные литеры: $Q_1 v ... v Q_n$. С помощью таких дизъюнктов можно представлять неполные сведения. Например, дизъюнкт *Рисует*(*Миша*)v*Рисует*(*Петя*) можно интерпретировать так: "хотя бы один из мальчиков Миша или Петя рисует", но не уточняется рисует только Миша, рисует ли только Петя или оба мальчика рисуют.

Если $m \geq 1$ и $n \geq 1$, дизъюнкт имеет самый общий вид и представляет условную дизъюнктивную информацию. Наличие такой дизъюнктивной информации позволяет выражать неполные знания о мире. Однако, формализовать рассуждения на основе таких знаний очень трудно. Поэтому в языках логического программирования, простых дедуктивных базах данных не используют такие дизъюнкты.

2.7.2. Хорновские дизъюнкты.

В языках логического программирования, подобных Прологу, ограничиваются **хорновскими дизъюнктами**, т. е. дизъюнктами, содержащими не более одного позитивного литерала. При $m \geq 1$ и $n=1$ дизъюнкт называется **точным хорновским дизъюнктом**, который имеет вид:

$P_1 \wedge ... \wedge P_m \rightarrow Q$.

Такой дизъюнкт осуществляет представление условной информации и может выступать в качестве правила в Пролог-программе. Точный дизъюнкт выражает некоторое правило: негативные литеры соответствуют *гипотезам*, а позитивная литера - *заключению*.

Если антецедент приведенной импликации не содержит литералов (m=0), то, очевидно, имеем дело с безусловной информацией. Такой дизъюнкт называется **унитарным позитивный дизъюнктом**. Возможны два случая.

- Литерал Q не содержит переменных. Тогда он представляет один конкретизированный литерал, например: *Рисует*(*Миша*). Такой дизъюнкт назовется **фактом**. Так выражаются факты в Пролог-программах, конкретизации отношений в реляционных базах данных.
- Литерал Q содержит переменные. Тогда он представляет более общую информацию о термах (переменные рассматриваются универсально квантифицированными).
- Негативные хорновские дизъюнкты.

Дизъюнкт, не содержащий ни одного позитивного литерала назовется **негативным**. Если в таком дизъюнкте один негативный литерал ¬P, то он представляет элементарную негативную информацию. Возможны два случая:

- Негативный литерал не содержит переменных. В этом случае он представляет собой элементарный негативный факт. Например, ¬*Рисует*(*Миша*).
- Негативный литерал содержит переменные. В этом случае он представляет негативную информацию о всех индивидных константах. В базах данных такой литерал должен обрабатываться как ограничение целостности.

Дизъюнкт, содержащий более одного негативного литерала, отражает более сложную негативную информацию. Например, ¬(*Летает*(x)^*Ползает*(x)). Интерпретация этого дизъюнкта такова: "ни один объект не может быть одновременно и летающим и ползающим".

В логическом программировании негативный хорновскии дизъюнкт представляется в виде вопроса, задаваемой системе вывода. Чаще позитивная информация, для которой сформулировано отрицание, обосновывается методом от противного.

Пустой дизъюнкт.

При m=0 и n=0 имеем пустой дизъюнкт, значение истинности которого всегда ложно, и поэтому никогда не представлен в базе данных.

В логическом программировании системы выводимости, использующие метод доказательства от противного, ориентированы на установлении противоречия путем получения пустого дизъюнкта.

Заметим, что в логическом программировании используется более выразительный язык, чем язык хорновских дизъюнктов. Например, разрешается внутри условных частей дизъюнктов использовать *оператор отрицания через конечную неудачу*. Кроме того могут включаться так называемые расширенные дизъюнкты, которые в антецеденте импликации могут содержать негативные литералы.

Особенность применения метода резолюции к хорновским дизъюнктам определяется тем, что правила в Прологе представляются именно в виде *хорновских дизъюнктов*, и эффективность метода резолюции в этом частном случае обеспечивается.

Задача вывода может быть сведена к проверке какой-то формулы, называемой целью, логически выводимой из фактов и правил. Для хорновских дизъюнктов метод резолюции по существу не претерпевает никаких изменений, только рассуждения ведутся в понятиях фактов, правил и цели (отрицания цели). Тогда выполнимость множества хорновских дизъюнктов может быть проверена с помощью следующего алгоритма

В множестве S выбираем факт P и какой-то дизъюнкт C из S, содержащий ¬P. Очевидно, будем иметь резольвенту, скажем, R. Заменяем исходное множество S на новое, представляющее логическую разность S с C и объединение с резольвентой:

$(S\setminus\{C\})\cup(\{R\})$.

На каждом этапе удаляется одна литера. По существу R - это дизъюнкт C, из которого удален символ ¬P. Отсюда следует, что как бы ни выбирались (т.е. начало алгоритма) P и C выполнение алгоритма конечно. Если число литер в исходном множестве S равно N, то цикл будет выполняться не более N раз.

В результате выполнения алгоритма возможны два варианта:

- либо порождается пустой дизъюнкт,
- либо получено новое множество S", не содержащее дизъюнктов, равных P и C. Поольку резольвенты получаются из дизъюнктов, принадлежащих S, т.е. резольвенты являются следствиями из S-дизъюнктов, вывод пустого дизъюнкта (всегда ложного) означает невыполнимость S.

Второй случай означает невозможность достижимости пустого дизъюнкта, а значит выполнимость S. Тогда множество дизъюнктов, полученных в результате выполнения алгоритма, называется **канонической моделью** S.

В заключение этой главы заметим, что системы логического программирования, подобные языку Пролог, пользуются языком хорновских дизъюнктов. Пролог-программа состоит из точных хорновских дизъюнктов. Интерпретатор Пролога является системой выводимости, которая в качестве правила вывода использует правило резолюций. Доказательства (обоснования вывода) осуществляется от противного. Чтобы установить, что формула $\exists y_1 ... y_K (P_1 \wedge ... \wedge P_m)$ является логическим следствием программы на Прологе, интерпретатор пытается доказать, что присоединение отрицания этой формулы к рассматриваемой базе знаний, которую обрабатывает Пролог-программа, приводит к противоречию.

Это эквивалентно тому, что добавление к программе хорновского дизъюнкта без консеквента $\neg P_1 v ... v \neg P_m$ (а он эквивалентен формуле $\neg(P_1 \wedge ... \wedge P_m)$) позволяет вывести пустой дизъюнкт.

Импликативная форма хорновского дизъюнкта, как отмечалось, соответствует правилу Пролог-программы, а конкретизированный позитивный литерал - факту. Тогда из следующих формул можно сформировать следующую элементарную Пролог-программу:

исходные формулы	*Пролог-программа*
пианист(X)→*музыкант*(X)	*музыкант*(X):-*пианист*(X).
пианист(*н_петров*).	*пианист*(*н_петров*).

На вопрос является ли Н. Петров музыкантом на основе метода резолюции будет сделан следующий вывод:

1. ¬*пианист*(X)∨*музыкант*(X)
2. *пианист*(*н_петров*)
3. ¬*музыкант*(*н_петров*)
4. *пианнст*(*н_петров*) - резольвента из 1 и 3.
5. □ из 2 и 4.

Ответ системы - да.

Для достижения большей выразительности языка, Допускают, как уже отмечалось, присутствие негативных выражений в антецедентах правил. С этой целью используют специальный предикат not. Эта конструкция использует *принцип отрицания через конечную неудачу*, который формулируется так: если задан позитивный литерал R, то предикат not(R) достигается и принимает значение истинности И тогда, когда не удается найти конечного доказательства для R на основе информации, имеющейся в рассматриваемой системе. В остальных случаях предикат not(R) принимает значение Л.

музыкант(X):-*пианист*(X), *not*(*аномалия*(X)).
аномалия(X) :-*нет_слуха*(X).
пианнст(*н_петров*).
?- *музыкант*(*н_петров*). (Вопрос к системе, т.е. доказываемая теорем).

Первое правило может быть интерпретировано так: "всякий X считается музыкантом, если он пианист (играет на рояли) и у него нет аномалий в виде отсутствия, в частности, музыкального слуха".

На поставленный вопрос система ответит да, поскольку значение (*аномалия*(X)) не удается вывести из имеющейся информации, но предикат *not*(*аномалия*(*н_петров*)) достигается и получает значение И.

Добавим к программе факт - *нет_слуха*(*д_федюшин*):

музыкант(X):-*пианист*(X), *not*(*аномалия*(X)).

аномалия(X) :-*нет_слуха*(X).

пианнст(*н_петров*).

нет_слуха(*д_федюшин*)

?- *музыкат*(*д_федюшин*).

Ответ системы: нет.

Предикат *not*(*аномалия*(*д_федкшин*)) потерпит неудачу, так как предикат *аномалия*(*д_федюшин*) будет достигнут. И, таким образом, будет отвергнуто утверждение *музыкант*(*д_федюшин*).

Упражнения.

Звездочками отмечены задания, для которых приведены Решения.

1. * Записать на языке предикатов:
   a) все студенты учатся;
   b) некоторые студенты отличники;
   c) для любого числа можно найти большее число;
   d) x+y=z
   e) всякий предмет обладает свойством А;
   f) нечто обладает свойством А ;
   g) всякий предмет не обладает свойством А;
   h) нечто не обладает свойством А;
   i) каждое рациональное число есть действительное число;
   j) некоторые рациональные числа не являются действительными.
2. * Попытайтесь объяснить, почему в упражнениях 1a и 1i использована импликация, а в 1j — конъюнкция.
3. * Записать на языке предикатов:
   a) детям до 16 лет D(x) и роботам R(x) входить B(x) запрещено;
   b) всем детям до 16 лет D(x) и роботам R(x) надлежит получить справки C(x).
4. * Записать на языке предикатов:
   a) всякое число, делящееся на 12, делится на 2, 4 и 6;
   b) каждый студент выполнил, по крайней мере, одну лабораторную работу;
   c) через две разные точки проходит единственная прямая.
5. Записать на языке предикатов:
   a) если нельзя решить задачу на данной ЭВМ, а задача важна для пользователя, то существует возможность разбить эту задачу на подзадачи или убедить пользователя, что решать задачу не имеет смысла;
   b) * каждый студент (C(x)) - спортсмен (S(x)) имеет какого-нибудь кумира y (B(x, y)) среди киноартистов (K(y));
   c) не всегда из того, что x лучше y, а y лучше z следует, что x лучше z,
   d) "Мой дядя самых честных правил,
      Когда не в шутку занемог,
      Он уважать себя заставил
      И лучше выдумать не мог".
6. * При каких условиях
   a) $\forall xP(x)\equiv\exists xP(x)$;
   b) $\exists xP(x)\equiv 0$, а $\forall xP(x)\equiv 1$.
7. Построить возможные варианты отрицаний предложений:
   a) все студены - отличники;
   b) есть студенты - отличники;
   c) существует число, которое больше всех остальных чисел.
8. * Этот ставший классическим пример, иллюстрирует дополнительные сложности, связанные с отрицанием: известно, что предложение "Нынешний король Франции лыс" не соответствует действительности. Как это записать на языке предикатов?
9. Придумать содержательные интерпретации формул:
   a) $\forall x(\exists y(P(x, y)\&Q(y)\rightarrow R(x, y))$;
   b) $\exists x(P(x)\rightarrow R(x))$;
   c) $\forall x(P(x)\hat{}R(x))$;
   d) $\forall x(P(x) \leftrightarrow \forall x(\exists yQ(x, y)\rightarrow R(x))$.
10. Доказать, что в общем случае разноименные кванторы нельзя переставлять местами (т.е. нельзя доказать, что $\forall x\exists yP(x, y) \equiv \exists y\forall xP(x, y)$).

11. Получить дизъюнкты для аксиом:
    a) * $\forall xP(x) \leftrightarrow \exists yQ(y)$.
    b) $\exists x(\forall yP(x, y) \rightarrow \exists z(Q(x, z) \rightarrow R(z))$.
    c) $\forall x(\forall y, z, v(R(x, y, z, v) \wedge L(y, z)$
    d) $\forall x, y, z(P(x) \wedge Q(x, y) \vee \neg R(z))$.
    e) $\exists x, y, z(P(x) \vee \neg Q(x, y) \rightarrow R(z) \vee M(y))$.
    f) $\forall xP(x) \wedge \exists xQ(x) \vee \forall x(R(x) \rightarrow P(x))$.
    g) * $\exists xP(x) \wedge \exists x R(x)$.
    h) $\forall xP(x) \wedge \forall xR(x)$.
    i) $\exists xP(x) \vee \exists xR(x)$.
    j) * $\forall xP(x) \vee \forall xR(x)$.
    k) $(\exists xP(x) \vee \forall yQ(y)) \wedge (\forall x, yR(x, y) \vee \exists yL(y))$.
12. Для данной системы аксиом доказать теорему,
    a) A_1. $\forall x, y(P(a, x, x) \vee P(b, y, y))$.
       Доказать теорему
       B_2. $\exists x \forall y \exists zP(x, y, z)$.
    b) * A_1. $\forall x(\neg P(x) \vee R(x) \vee Q(g(x)))$;
       A_2. $\forall x(Rx) \rightarrow L(x) \wedge M(x))$;
       A_3. $\forall x(q(x) \rightarrow L(x) \wedge D(x))$,
       A_4. $\forall x(P(\varphi(x)) \vee R(\eta(x)))$.
       Доказать теорему:
       B_5. $\exists x, y((L(x) \wedge M(x)) \vee (D(y) \wedge L(y)))$.
    c) * A_1. $\forall x(\exists y(S(x, y) \wedge M(y) \rightarrow \exists y(L(y) \wedge R(x, y)))$.
       Доказать теорему:
       B_2. $\neg \exists xL(x) \rightarrow \forall x, y(S(x, y) \rightarrow \neg M(y))$.
    d) A_1. $\forall y(S(y) \rightarrow C(y))$.
       Доказать теорему:
       B_2. $\forall x(\exists y(S(y) \wedge R(x, y)) \rightarrow \exists z(C(z) \wedge R(x, z)))$.
    e) * A_1. $\forall x(P(x) \rightarrow P(f(x)))$
       Доказать теорему:
       B_2. $\exists x \neg P(x)$.
    f) A_1. $\forall xB(x)$;
       A_2. $\forall x(B(x) \rightarrow C(x))$;
       A_3. $C(x) \rightarrow D(b)$;
       A_4. $\neg \exists x(D(x) \wedge E(y))$.
       Доказать теорему:
       B_5. $\forall x, y(C(x) \wedge \neg E(y))$.
13. Для упражнений п.2 преобразовать теоремы в вопросы и получить ответы с помощью метода резолюции.

14. Получить ответ с помощью метода резолюции:
    a)
        1. Все студенты — люди.
        2. Петя - студент.

        Вопрос: Кто человек?
    b)
        1. Все, что обладает свойством P, имеет свойство R.
        2. Все, что обладает свойством R, имеет свойство Q.

        Вопрос: Существует ли нечто, что обладает свойством P или обладает свойством Q?
    c)
        1. Если x есть часть y и y есть часть z, то x есть часть z.
        2. Палец есть часть кисти руки.
        3. Кисть есть часть руки.
    d) Вопрос: Частью кого является рука ?
    e) *
        1. Все отличники не имеют "хвостов".
        2. Некоторые двоечники имеют "хвосты".

        Вопрос: Существует ли двоечник, который не является отличником ?
    f)
        1. Кто ходит в гости по утрам, тот поступает мудро.
        2. Если у кого угодно есть воздушный шарик, то он ходит в гости по утрам.
        3. У Пятачка есть воздушный шарик.

        Вопрос: Кто поступает мудро ?
    g)
        1. Если робот обработал деталь, то ее забирает штабелер.
        2. Если деталь поступила на обработку, то ее обработает робот.
        3. Если человеку нужна деталь, то она поступит на обработку
        4. Человеку нужна втулка.

        Вопрос: Что заберет штабелер ?
    h)
        1. Если пассажир сел в самолет, на который ему удалось купить билет, и этот самолет взлетел, то пассажир думает, что данный самолет разобьется.
        2. Если пассажир не сел ни на какой самолет или самолет не взлетел, то безопасность пассажира гарантируется.
        3. Безопасность пассажира Васи не гарантируется.

        Вопрос: Кто думает, что самолет, на который ему удалось купить билет, разобьется ?
    i) *
        1. Всякая большая программа содержит подпрограммы.
        2. Для всех программ существуют программы такие, что если программа небольшая, неструктурированная и не содержит подпрограмм, то программа и подпрограмма все равно не соизмеримы по сложности.
        3. Если программы и подпрограммы соизмеримы по сложности, то значит подпрограммы структурные или большие.
        4. Существуют программы, соизмеримые по сложности с подпрограммами.
        5. Программа pI неструктурная.
        6. Если программа содержит подпрограммы, то они находятся в одной библиотеке.

        Вопрос1: Какая программа большая ?

        Вопрос2: Какая подпрограмма находится в одной библиотеке с pI ?
    j) Придумать новые вопросы и получить ответы для задачи 4,з.

15. Доказать теорему и получить ответ ОТВ.
    a) * Аксиомы:
        1. Некоторые пациенты любят своих докторов.
        2. Ни один пациент не любит знахаря.
        Доказать терему:
        3. Никакой доктор не является знахарем.
    b) Аксиомы:
        1. Если x ребенок (дитя) для y, то y родитель x.
        2. Каждый ребенок является ребенком своего родителя.
        Вопрос:
        3. Для любого л кто является его родителем?
    c) Аксиомы:
        1. Каждый объект a, находящийся в некоторой точке y в каком-то состоянии z, может быть перемещен в другую точку (без промежуточных точек) под некоторым воздействием.
        2. Конкретный объект находится в фиксированной точке в определенном состоянии.
        Вопрос:
        3. 3. Каково действия по перемещению объекта из точки a в точку b?