

МГТУ ФН-12 МГТУ ФН-12 МГТУ

Московский государственный технический университет
имени Н.Э. Баумана

Факультет «Фундаментальные науки»
Кафедра «Математическое моделирование»

А.Н. Канатников

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Конспект лекций

Для студентов кафедры ИУ9

Москва
2010

МГТУ ФН-12 МГТУ ФН-12 МГТУ

3. АЛГЕБРА ПРЕДИКАТОВ

3.1. Предикаты и кванторы

Ограниченность исчисления высказываний. Переменные высказывания. Субъект и предикат переменного высказывания. Понятие предиката в математической логике. Арность предиката. Контуры будущей теории. Кванторы как символы агрегирования. Выражения двух видов.

Построенное исчисление высказываний в реальности оказывается довольно бедной теорией. Оно позволяет строить и анализировать различные комбинации высказываний. Однако в математике встречаются предложения, которые, вообще говоря, не являются высказываниями. Например, нельзя сказать, истинно ли утверждение „число x положительное“. Ответ зависит от того, какое это число. Здесь мы сталкиваемся с уточнением того, что такое суждение. В каждом суждении присутствуют два компонента — субъект и предикат. Субъект — это понятие, отражающее предмет, о котором идет речь. В предложении субъект связывается с подлежащим. Предикат же можно уподобить сказуемому, поскольку этот структурный элемент выражает свойства, приписываемые предметам или отрицаемые у них. В суждениях „Роза красная“, „Гитара семиструнная“ субъектами являются „роза“ и „гитара“, а предикатами — „красная“ (все то, что называется красным) и „семиструнная“ (все то, что называется семиструнным).

В математике этой трактовке суждения отвечает „высказывание с параметрами“, т.е. суждение, которое становится высказыванием при конкретизации параметров. Такие суждения называют предикатами. С математической точки зрения предикат можно интерпретировать как некое отображение, которое каждому набору входящих в него параметров ставит в соответствие высказывание. Количество участвующих в предикате параметров называется его **арностью**.

Понятие предиката не вкладывается в язык исчисления высказываний. Требуется построение более широкой теории, требующей и расширения языка теории. Что должен охватывать такой язык помимо высказываний и логических связей?

Во-первых, если в формулах алгебры высказываний встречаются переменные, областью действия которых является множество высказываний, то в формулах любой математической теории есть переменные, область действия которых связана не с высказываниями, а с объектами, изучаемыми в этой теории. В математическом анализе переменные числовые, в геометрии переменные обозначают точки, прямые, плоскости. Такие переменные называют **предметными**, а область их действия — **предметной областью**. В теории могут участвовать предметные переменные разного рода, поскольку описывают объекты разных классов (например, точки и прямые в геометрии).

Во-вторых, в ряде математических теорий есть установленные символы для обозначения некоторых стандартных объектов (например, число π). В отличие от переменных эти символы имеют конкретное значение. Их называют константами.

В третьих, в каждой математической теории используются функции, причем разные, к которым можно отнести и операции над объектами математической теории. Функции можно рассматривать двояко: как отображение и как связь между переменными. С точки зрения формальной записи функция — это символ, который исходя из некоторой совокупности переменных формирует новую переменную (точнее, новый объект, имеющий неопределенное значение).

Наконец, в четвертых, в этом языке должны быть отражены и такие символы, как кванторы. Кванторы — специальные символы (или фразеологические штампы), которые позволяют

из неопределенного предложения построить правильное высказывание, истинность которого устанавливается однозначно. Кванторы можно рассматривать как символы агрегирования, которые из совокупности возможных значений формируют одно значение. Таковы символы предела, интеграла, суммы. В логике используют два квантора: всеобщности $\forall$ и существования $\exists$. Запись $\forall xP(x)$ формирует высказывание, являющееся истинным, если предикат $P(x)$ является истинными для любого возможного значения предметной переменной x . Запись $\exists xP(x)$ формирует высказывание, истинное, если предикат $P(x)$ имеет значение истины хотя бы для одного возможного значения предметной переменной x .

Введение в язык символов агрегирования приводит к тому, что вхождение переменных в формулы может играть две разных роли. В формуле $x + y$ переменным x и y можно присвоить конкретные значения, в результате чего конкретное значение получит все выражение. В формуле $\forall xP(x)$ переменной x нельзя присвоить конкретное значение, поскольку формула подразумевает использование целого множества значений переменной, а не отдельно взятого значения (поэтому и квантор — символ агрегирования). В первом случае говорят о **свободном вхождении переменной**, а во втором — о **связанном вхождении**. Одна и та же переменная в выражение может входить и свободно, и связано. Например, в выражении $n + \sum_{n=1}^3 a_n$ переменная n входит свободно (первое вхождение) и связано (второе и третье).

В дальнейшем изложении мы не будем использовать предметных символов агрегирования, ограничившись только двумя кванторами, так как предметные символы агрегирования усложняют язык, но не добавляют чего-то существенного в формальную теорию как таковую. Отметим, что „за бортом“ остаются обозначения множеств и связи множеств и элементов. Если в формальной теории нет средств для обозначения множеств, ее называют теорией 1-го порядка. При введении множеств отдельных элементов получаем теорию 2-го порядка, множества множеств приводят к теории 3-го порядка. И так далее.

Отметим, что новая теория будет содержать выражения двух видов. Первые не означают какое-то высказывание, они описывают лишь действия, выполняемые над объектами теории. Например, выражение $x + y$ играет роль новой предметной переменной, значения которой однозначно определяются значениями x и y . Напротив, выражение $x + y = 0$ выражает суждение, которое может быть истинным или ложным при заданных значениях неизвестных. Разделяя два типа выражений, первые будем называть **термами**, а вторые **формулами**. Таким образом, значением терма является некоторый объект теории (например, число), в то время как значением формулы является высказывание.

3.2. Логико-математические языки

Язык как совокупность четырех множеств (V, C, F, P) . Предметные переменные, константы и их сортность. Функциональные и предикатные символы, их тип. Предикатные символы арности 0 как пропозициональные переменные. Индуктивное построение термов и формул. Функциональная сложность терма, логическая сложность формулы. Выражения. Правильные формулы: $A(x)$, $B(x, y)$, C , $(\forall xA(x))$. Комментарии по вариантам построения логико-математических языков. Односортные языки. Свободные и связанные вхождения переменных. Предложения (замкнутые термы и формулы). Соглашения о сокращенной записи формул.

Под логико-математическим языком мы будем понимать совокупность (V, C, F, P) из четырех не более чем счетных множеств, элементы которых называют **символами**. Первое множество — это множество V **предметных переменных**. Предметные переменные могут иметь разные **сорты**. Конкретно мы считаем, что есть еще пятое множество S — **множество сортов** и задано отображение $\text{sort}_V: V \rightarrow S$. Значение отображения sort_V на переменной $x \in V$ и есть сорт этой переменной. Каждому сорту $\pi \in S$ соответствует множество V_π предметных переменных этого сорта, являющееся полным прообразом $\text{sort}_V^{-1}(\pi)$ элемента $\pi \in S$.

Элементы множества C называются **константами**. Каждая константа относится к определенному сорту $\pi \in S$, т.е. задано отображение $\text{sort}_C: C \rightarrow S$ множества констант в множество сортов.

Элементы множества F — это **функциональные символы**, каждый из которых характеризуется своим типом, определяемым кортежем $(\pi_0, \pi_1, \dots, \pi_n) \in S^{n+1}$ сортов (целое число $n > 0$ называется **арностью функционального символа**).

Элементы множества P — это **предикатные символы**, каждый из которых характеризуется своим типом, определяемым кортежем сортов $(\pi_1, \dots, \pi_n) \in S^n$ (и в этом случае $n \geq 0$ — **арность предикатного символа**).

Названия элементов четырех множеств указывают на роль, которые они будут играть в содержательной интерпретации языка. Однако этот содержательный смысл не играет роли в формальной теории. Главное, каким образом из символов составляются правильные слова языка.

В логико-математическом языке два вида правильных слов. Это термы и формулы. Начнем с индуктивного определения термов.

База индукции. Любая переменная или константа сорта π является термом сорта π (элементарные термы).

Шаг индукции. Если $t_1, t_2, \dots, t_n$ — термы сортов $\pi_1, \dots, \pi_n$ соответственно, f — функциональный символ типа $(\pi_0, \pi_1, \dots, \pi_n)$, то $f(t_1, t_2, \dots, t_n)$ — терм сорта π_0 .

Из этого определения нетрудно понять, что термы служат для записи выражений в предметной области (например, многочленов в полях).

Функциональной сложностью терма называется количество функциональных символов в нем. Индуктивно эту характеристику можно определить так: сложность элементарного терма равна 0; сложность терма $f(t_1, t_2, \dots, t_n)$ равна сумме сложностей аргументов плюс один, т.е.

$$|f(t_1, t_2, \dots, t_n)| = |t_1| + |t_2| + \dots + |t_n| + 1,$$

где знак модуля обозначает функциональную сложность терма.

Подобное индуктивное определение может использоваться не только для определения понятий, но и для доказательства различных утверждений. Например, пусть все элементарные термы обладают свойством P ; из того, что термы $t_1, t_2, \dots, t_n$ обладают свойством P , следует, что терм $f(t_1, t_2, \dots, t_n)$ также обладает свойством P . Тогда все термы обладают свойством P .

Определим теперь формулы.

База индукции. Если $t_1, t_2, \dots, t_n$ — термы сортов $\pi_1, \dots, \pi_n$ соответственно, p — предикатный символ типа $(\pi_1, \dots, \pi_n)$, то $p(t_1, t_2, \dots, t_n)$ — формула (такую формулу будем называть **элементарной**).

Шаг индукции. Если X_1, X_2 — формулы, то $(X_1 \rightarrow X_2)$, $(X_1 \wedge X_2)$, $(X_1 \vee X_2)$, $\neg X_1$ — формулы. Если X — формула, x — предметная переменная, то $\forall x X$ и $\exists x X$ — формулы.

В то время как термы представляют собой записи последовательностей операций над объектами предметной области, формулы обозначают те или иные утверждения о свойствах этих объектов. Совокупность всех термов в рассматриваемом языке обозначим Tm , совокупность всех формул — Fm , а совокупность и тех, и других — Expr . Элементы Expr будем называть **выражениями**.

Отметим, что допускаются предикатные символы арности 0, которые являются элементарными формулами. Множество таких символов в сочетании с логическими связками образует подмножество языка, эквивалентное языку алгебры высказываний. Поэтому предикатные символы арности 0 можно рассматривать как пропозициональные переменные.

Для формул может быть введена характеристика, называемая **логической сложностью**, аналогичная функциональной сложности термов, но определяемая количеством предикатных символов в формуле.

Замечание. 1. На самом деле понятие логико-математического языка можно вводить различными способами, не меняя содержательной сути. Так, предикатные символы можно

рассматривать как особый род функциональных символов, для которых сорт π_0 есть „высказывание“. Аналогично можно было бы отказаться от переменных языка, рассматривая их как функциональные символы арности 0. В результате дело можно было бы свести к двум множествам: S и F .

Отметим также, что множество V на самом деле можно трактовать как семейство множеств, в котором каждое множество — это множество переменных одного сорта. Чтобы обозначить сорт переменной, достаточно указать множество, которому эта переменная принадлежит.

Выбор конкретного варианта определения диктуется субъективными факторами, а также тем, что должно присутствовать нечто, трактуемое как сорт высказываний.

2. Совокупность всех переменных, констант, функциональных и предикатных символов плюс логические связки, кванторы, скобки и запятая составляет алфавит логико-математического языка. В данном случае алфавит оказывается счетным множеством. Однако нетрудно дело свести к конечному алфавиту, если считать, что элементы логико-математического языка на самом деле являются словами другого языка (языка нижнего уровня). Так, для всех натуральных чисел можно получить обозначения с помощью двух символов 0 и 1, используя позиционную форму записи. Однако этот механизм находится за пределами интересов математической логики. Поэтому мы и считаем идентификаторы переменных, функций и т.п. простейшими неделимыми элементами языка — символами. #

Среди логико-математических языков выделим **односортные языки**, в которых все предметные переменные имеют один сорт. Таков язык, например, арифметики. Напротив, язык линейной алгебры является двусортным (числа и векторы).

В формулах $\forall x X$ или $\exists x X$ первый символ называется **квантором** (далее, если несущественно, какой именно квантор, мы будем использовать обозначение ∇). Комбинация из квантора и переменной называется **кванторной приставкой**, а формула X , следующая за кванторной приставкой — **областью действия квантора**. Допускается, что переменная кванторной приставки (**кванторная переменная**) вообще не имеет вхождений в области действия этой приставки.

Появление кванторов приводит к тому, что предметные переменные формулы по своему положению могут иметь разные свойства. В языке высказываний была введена такая операция, как подстановка вместо переменных других переменных или формул. Однако при наличии кванторов произвольная подстановка может привести к неправильной формуле. Например, в формуле $\forall x (x > 0)$ нельзя переменную x заменить произвольной формулой: слово $\forall (x + 1) ((x + 1) > 0)$ не является формулой, потому что по индуктивному правилу образования формул сразу за квантором должна следовать переменная, а не служебный символ или константа. Содержательный смысл формулы при такой подстановке также теряется.

Будем говорить, что вхождение предметной переменной x в формулу X является **связанным**, если оно есть часть кванторной приставки или находится в области действия кванторной приставки с той же переменной. В противном случае вхождение переменной называется **свободным**. В терме все вхождения каждой переменной свободные.

В одной и той же формуле переменная x может иметь и свободные, и связанные вхождения. Например, в формуле $p_1(x) \vee \forall x p_2(x)$ три вхождения переменной x , причем первое вхождение свободное, а два других связанные. Если переменная x имеет свободные вхождения в формулу X , то ее называют **свободной переменной** этой формулы. Отметим, что определенное вхождение переменной может быть связанным в формуле и свободным в некоторой подформуле. Например, предикатный символ, в один из аргументов которого входит переменная, является подформулой, а в элементарной формуле вхождение любой переменной свободное. Еще один пример. В формуле

$$(\forall x p(x, y) \wedge q(x)) \rightarrow (\exists x r(x, x))$$

полужирным шрифтом помечены связанные вхождения переменных, а светлым — свободные.

Свободные переменные, входящие в терм или формулу, называются **параметрами** этого терма или формулы. Если формула не имеет свободных переменных, то она называется **замкнутой** или **предложением**.

При записи формул будем использовать сокращения скобок на основе приоритета операций. Наивысшим приоритетом обладают кванторные приставки. Затем идет отрицание, далее дизъюнкция и конъюнкция. Наинизшим приоритетом обладает импликация. Мы будем также использовать сокращение $(X \sim Y)$ для формулы $((X \rightarrow Y) \wedge (Y \rightarrow X))$, причем символу $\sim$ (эквиваленция) припишем самый низкий приоритет (ниже, чем у импликации). Кроме того, некоторые из функциональных символов арности 2, будем употреблять в инфиксной форме (например, символы алгебраических операций). Все эти соглашения используются для более удобной записи и с теоретическими построениями не связаны. От подобных соглашений всегда можно избавиться, модифицировав соответствующим образом терм или формулу. Разумеется, эти элементы можно было бы включить в формальный язык, но это в теоретическом плане ничего существенного не добавит, хотя заметно усложнит изложение.

3.3. Переименования и подстановки

Роль свободных и связанных переменных. Переименование. Коллизия переменных при переименовании. Отношение конгруэнтности ($\approx$). Подстановка, свободная для данного выражения. Сохранение конгруэнтности при подстановке. Свойство чистоты переменных. Лемма о чистоте переменных.

С понятием связанной переменной мы встречаемся уже в курсе математического анализа. Обозначения предела, определенного интеграла дают примеры связывания переменных. Из того же курса известно, что связанную переменную можно изменять, не меняя смысла выражения. Однако это в общем верное правило может приводить к неверным результатам. Например, если в выражении $\lim_{x \rightarrow 0} \cos(x^2 + y^2)$ заменить связанную переменную x переменной y , получим формулу $\lim_{y \rightarrow 0} \cos(y^2 + y^2)$, имеющую совсем другой смысл. В этом случае говорят, что при подстановке произошла коллизия переменных.

Одновременная замена переменной x в кванторной приставке ∇x и всех свободных вхождений переменной x в области действия этой приставки на переменную y того же сорта называется **переименованием**. **Коллизия переменных при переименовании** в формуле $\nabla x X$ — ситуация, когда при переименовании переменной x в переменную y в X или в одной из ее подформул есть переменная, которая из свободной превращается в связанную. Например, в формуле $\forall x p(x, y, z)$ переменные y и z являются свободными. Но при переименовании x в y получим $\forall y p(y, y, z)$, где только одна переменная z осталась свободной. Может быть и другая ситуация. В формуле $\forall x p_1(x, (\forall y p_2(x, y)), z)$ оба вхождения переменной y связанные. Переменная x в формулу входит связанно (раз есть квантор), но в подформулу $\forall y p_2(x, y)$ эта переменная уже входит свободно. Но переименование x в y переводит y в разряд связанной переменной в этой подформуле: $\forall y p_1(y, (\forall y p_2(y, y)), z)$. В этом случае также произошла коллизия переменных.

Можно сказать по-другому. Назовем переменную x в формуле Z **свободной переменной** для переменной y , если никакие вхождения y в Z , являющиеся свободными для Z или одной из ее подформул, не попадают в область действия кванторных приставок с переменной x . Коллизия переменных при переименовании — попытка заменить кванторную переменную x на y в то время как x не является свободной для y .

Мы будем говорить, что две формулы X и X' **конгруэнтны** (являются вариантами одна другой, обозначение $X \approx X'$), если одна получена из другой с помощью правильного, т.е. без коллизии переменных, переименования. Точное определение этого понятия можно дать индукцией по построению формулы:

- 1) элементарная формула конгруэнтна только себе самой;

* Напомним, что символ ∇ обозначает любую из двух кванторных приставок.

2) если $Y \approx Y'$, $Z \approx Z'$, то $\neg Y \approx \neg Y'$ и $Y \nabla Z \approx Y' \nabla Z'$, где ∇ — одна из трех двуместных логических связей;

3) если $X = \nabla x Y$, $X' = \nabla y Z$, то $X \approx X'$, если x и y одного сорта и для любой переменной z того же сорта, вообще не входящей ни в Y , ни Z , имеем $Y_z^x \approx Z_z^y$, где W_u^v есть результат замены в W всех свободных вхождений переменной v на переменную u .

Отметим, что конгруэнтность — это отношение эквивалентности (симметричное, рефлексивное и транзитивное). Таким образом, множество всех формул разделяется на классы попарно конгруэнтных формул. Хотя формально конгруэнтные формулы различаются, нет оснований такое различие учитывать. С математической точки зрения здесь осуществляется переход от понятия „формула“ к более общему понятию „класс конгруэнтных формул“.

Под подстановкой θ мы будем понимать символическую запись вида

$$\theta = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ t_1 & t_2 & \dots & t_n \end{pmatrix},$$

в которой $x_1, x_2, \dots, x_n$ — конечный набор (предметных) переменных, а $t_1, t_2, \dots, t_n$ — набор термов, причем сорт терма t_i совпадает с сортом переменной x_i . Такую запись мы всегда можем трактовать как запись отображения, определенного на конечном подмножестве множества всех предметных переменных языка, в множество T_m всех термов языка, которое сохраняет сортность термов. Допускается, что приведенная таблица может быть пустой.

Далее через $T\theta$ будем обозначать результат подстановки θ в выражение (т.е. терм или формулу) T , т.е. результат замены в T всех свободных вхождений каждой переменной x_i из подстановки θ соответствующим термом t_i . Отметим, что при этом некоторые переменные x_i могут не входить свободно в T . В этом случае в отношении x_i не происходит никаких замен, соответствующий столбец в подстановке θ можно опустить.

В алгебре высказываний подстановка — один из способов преобразования формулы в эквивалентную. При этом на подстановку не накладывалось никаких ограничений. Ситуация меняется с появлением кванторов. При подстановке в формуле с кванторами смысл формулы может заметно измениться. Поэтому не все подстановки пригодны с точки зрения логики. Например, формула^{**} $x + y = z$ с натуральными переменными x, y, z есть предикат от трех переменных. Формула $\exists y (x + y = z)$ есть предикат уже от двух переменных x и z , логическим эквивалентом которого является формула $x < z$. Заменяем в этой формуле свободную переменную x термом $x \cdot y$, надеясь на то, что получим предикат $x \cdot y < z$. Однако формула $\exists y (x \cdot y + y = z)$ имеет совсем другой смысл. Здесь возникла та же проблема, что и при переименовании — коллизия переменных: при замене переменной x термом $x \cdot y$ переменная y , входящая в терм свободно, оказалась в области действия кванторной приставки и стала связанной. Чтобы избежать искажения, достаточно было предварительно „убрать“ из формулы связанную переменную y с помощью переименования а уж затем выполнить подстановку: $\exists u (x \cdot y + u = z)$. При таком порядке действий действительно получим предикат $x \cdot y < z$.

Подстановку θ назовем **свободной для выражения** T , если для любой переменной x_i из области определения $\text{dom } \theta$ подстановки θ (т.е. встречающейся в первой строке матрицы), никакое свободное вхождение x_i в T не попадает в область действия никаких кванторных приставок с переменными, свободно входящими в $t_i = \theta(x_i)$.

В частности, если ни один из параметров формул $\theta(x)$ не является кванторной переменной в T , то θ свободна для T . Это будет так в случае, когда каждая из формул $\vartheta(x)$ вовсе не имеет параметров, т.е. является замкнутой. Такая подстановка называется **константной**.

Если подстановка θ не является свободной для T , то, как показывает рассмотренный выше пример, можно, не меняя логической сути формулы, выполнить переименование переменных

^{**} Отмечу, что любое алгебраическое уравнение является не термом, а формулой: его можно рассматривать как утверждение, касающееся указанных переменных, при этом решение уравнения представляет собой выявление области истинности этой формулы.

в T , переходя к конгруэнтной формуле T' , для которой θ уже будет свободной. Этот подход позволяет обеспечить универсальность подстановки, но, правда, пожертвовав однозначностью: результат будет лишь с точностью до когруэнтности.

Лемма 3.1. Если $X' \approx X''$ и переменная u входит в X' свободно, то и в X'' переменная u входит свободно.

◀ Это интуитивно ясное утверждение (переименование не затрагивает свободных переменных) строго доказывается методом индукции по построению формулы. Утверждение очевидно для элементарных формул (конгруэнтность в этом случае равносильна совпадению). Если $X' = U' \vee W'$, $X'' = U'' \vee W''$, то конгруэнтность $X' \approx X''$ означает, что $U' \approx U''$ и $W' \approx W''$. Если, например, u входит свободно в U' , то, согласно индуктивному предположению, u входит свободно и в U'' , а значит, и в $X'' = U'' \vee W''$. Аналогичны рассуждения для остальных логических связок, включая отрицание.

Пусть $X' = \nabla x U'$, $x'' = \nabla y U''$, где ∇ — один из кванторов. Условие $X' \approx X''$ в данном случае означает, что для переменной z , не входящей в U' и U'' , имеем $(U')_z^x \approx (U'')_z^y$. Отметим, что в этом случае $u \neq x$, поскольку x не входит свободно в X' . Следовательно, u входит свободно в $(U')_z^x$. По индуктивному предположению u входит свободно в $(U'')_z^y$, откуда вытекает, что $u \neq y$, так как y не входит свободно в $(U'')_z^y$. Но тогда u входит свободно и в $X'' = \nabla y U''$. ▶

Теорема 3.1. Если $X' \approx X''$, а θ свободна для X' и X'' , то $X'\theta \approx X''\theta$.

◀ Доказательство базируется на индукции по построению формулы и на индуктивном определении конгруэнтности. Если $X' \approx X''$, то формулы X либо обе элементарные, либо обе получены с помощью логической связки, либо обе построены присоединением кванторной приставки. В первом случае X' и X'' совпадают, а значит, формулы $X'\theta$ и $X''\theta$ также элементарны и совпадают, т.е. $X'\theta \approx X''\theta$.

Пусть, например, $X' = U' \vee W'$ и $X'' = U'' \vee W''$, причем $U' \approx U''$, $W' \approx W''$. По индуктивному предположению $U'\theta \approx U''\theta$ и $W'\theta \approx W''\theta$. Следовательно, $U'\theta \vee W'\theta \approx U''\theta \vee W''\theta$ и

$$X'\theta = (U' \vee W')\theta = U'\theta \vee W'\theta \approx U''\theta \vee W''\theta = (U'' \vee W'')\theta = X''\theta.$$

Аналогичны рассуждения для других логических связок.

Пусть $X' = \nabla x U'$, $X'' = \nabla y U''$. Выберем переменную z , не входящую ни в формулы U' , U'' , ни в формулы подстановки θ . Тогда, согласно условию $X' \approx X''$, имеем $(U')_z^x \approx (U'')_z^y$. Согласно индуктивному предположению, $(U')_z^x \theta \approx (U'')_z^y \theta$. Отметим, что переменные x и y не входят свободно в формулы $(U')_z^x$ и $(U'')_z^y$, так как эти формулы конгруэнтны, x не входит свободно в первую из этих формул, y — во вторую. Рассмотрим подстановку $\bar{\theta}$, полученную из θ удалением столбцов с переменными x и y , если они есть. Применение подстановок θ и $\bar{\theta}$ к формулам $(U')_z^x$ и $(U'')_z^y$ дает один и тот же результат, поскольку переменные x и y не входят свободно в эти формулы. Следовательно, $(U')_z^x \bar{\theta} \approx (U'')_z^y \bar{\theta}$. Так как $(U')_z^x \bar{\theta} = (U'\bar{\theta})_z^x$ и $(U'')_z^y \bar{\theta} = (U''\bar{\theta})_z^y$, то $(U'\bar{\theta})_z^x \approx (U''\bar{\theta})_z^y$. В соответствии с определением конгруэнтности заключаем, что $\nabla x(U'\bar{\theta}) \approx \nabla y(U''\bar{\theta})$. Наконец, с учетом равенств $\nabla x(U'\bar{\theta}) = (\nabla x U')\bar{\theta} = (\nabla x U')\theta$ и $\nabla y(U''\bar{\theta}) = (\nabla y U'')\bar{\theta} = (\nabla y U'')\theta$ заключаем, что $X'\theta = (\nabla x U')\theta = (\nabla y U'')\theta = X''\theta$. ▶

Будем говорить, что формула X обладает **свойством чистоты переменных**, если, во-первых, никакая переменная кванторной приставки не входит в X свободно, и во-вторых, разные кванторные приставки содержат разные переменные.

Следующая лемма позволяет утверждать, что в любой формуле T можно устроить переименование, в результате которого данная подстановка θ станет свободной для переименованной формулы.

Лемма 3.2 (о чистоте переменных). Для любой формулы X и любого множества $S \subset V$ переменных существует формула Y , когруэнтная X , обладающая свойством чистоты переменных, в которой нет связанных переменных, принадлежащих S .

◀ Доказательство проводится индукцией по построению формулы. Для элементарных формул утверждение тривиально, поскольку такие формулы вообще не имеют связанных переменных. Пусть задано множество S и формула $X = Y \nabla Z$, где ∇ — одна из двуместных логических связок. Предположим, что относительно формул Y и Z уже доказано утверждение леммы с произвольным множеством S . Пусть S_X — множество всех переменных (и свободных, и связанных), входящих в X . Существует формула Y' , конгруэнтная Y , обладающая свойством чистоты переменных и не имеющая связанных переменных из множества $S \cup S_X$. Обозначим через S_Y множество всех переменных, входящих в Y' . Существует формула X' , конгруэнтная X , обладающая свойством чистоты переменных и не имеющая связанных переменных из множества $S \cup S_Y$. Формула $X' \nabla Y'$ обладает свойством чистоты переменных (любая кванторная переменная, например, в X' не совпадает со „своими“ переменными, т.е. другими кванторными переменными формулы X' , а по построению и с „чужими“ переменными, составляющими множество S_Y). Кроме того, в построенной формуле нет связанных переменных, попадающих в S . Для одноместной связки $\neg$ утверждение теоремы очевидно.

Пусть $X = \nabla x Y$, где ∇ — один из кванторов. Сперва надо „вывести“ новую кванторную переменную за пределы S . Для этого выбираем новую переменную u , не принадлежащую S и не входящую никаким образом в Y . Подстановка $\left(\begin{smallmatrix} x \\ u \end{smallmatrix}\right)$ свободна и мы получаем формулу $\nabla u Y_u^x$, конгруэнтную X . Если мы теперь заменим Y_u^x конгруэнтной формулой Y' со свойством чистоты переменных и без переменных из S , то получим конгруэнтную Y формулу $\nabla u Y'$ со свойством чистоты переменных и без переменных из S . ▶

В соответствии с доказанной леммой, чтобы выполнить в X правильную подстановку θ , необходимо переименовать X в X' так, что среди переменных X' не будут встречаться переменные термов $\theta(x)$. Тогда подстановка θ будет свободной для X' , и мы можем реализовать эту подстановку.

3.4. Семантика логико-математического языка

Интерпретация логико-математического языка. Предметная область. Интерпретация предметных констант. Интерпретация функциональных и предикатных символов. Интерпретация логических связок и кванторов. Расширение языка введением символов реальных объектов. Оценка и оцененные выражения.

Сами по себе термы и формулы логико-математического языка ничего не значат. Это лишь средство для обозначения или записи реальных объектов. Чтобы термы и формулы приобрели какой-то содержательный смысл, необходимо связать их с реальными объектами из какой-либо области математики, т.е. определить **интерпретацию логико-математического языка**, или построить **модель языка**. Для этого мы должны предусмотреть несколько вещей.

Во-первых, необходимо придать смысл предметным переменным, определив их возможные значения. Мы назовем **носителем** (или **предметной областью**) языка Ω отображение, которое каждому сорту $\pi \in S$ ставит в соответствие некоторое непустое множество D_π . Множество D_π представляет собой носитель сорта π или, другими словами, область изменения переменных сорта π .

Во-вторых, следует придать смысл предметным константам. Естественно каждую константу $c \in C$ сорта π связать с некоторым элементом $\tilde{c} \in D_\pi$ из носителя сорта π , т.е. задать отображение $\tilde{C}: C \rightarrow \bigcup_{\pi \in S} D_\pi$, сохраняющее сортность.

В-третьих, нужно каждому функциональному символу f типа $(\pi_0, \dots, \pi_n)$ поставить в соответствие конкретное отображение $\tilde{f}: D_{\pi_1} \times \dots \times D_{\pi_n} \rightarrow D_{\pi_0}$, т.е. задать некоторое отображение $\tilde{F}: f \mapsto \tilde{f}$.

В-четвертых, следует каждому предикатному символу p типа $(\pi_1, \dots, \pi_n)$ поставить в соответствие n -местный предикат того же вида, т.е. отображение, которое каждому элементу

множества $D_{\pi_1} \times \dots \times D_{\pi_n}$ ставит в соответствие высказывание. Впрочем, поскольку нас интересует лишь истинность формул, можно ограничиться булевым вариантом и символу p поставить в соответствие отображение $\hat{p}: D_{\pi_1} \times \dots \times D_{\pi_n} \rightarrow \mathbb{B}$. Отдельно упомянем предикатные символы арности 0. Это пропозициональные переменные, которые в случае булевой интерпретации можно заменить двумя булевыми константами.

В-пятых, следует придать смысл всем логическим связкам и кванторам. Мы будем считать, что они имеют общепринятый смысл и что их интерпретация фиксирована.

Все сказанное означает, что интерпретация — предписание, которое символам языка ставит в соответствие настоящие объекты. Говорят, что модель языка определяет его **семантику**.

Если задана интерпретация M языка Ω , то каждый терм сорта π определяет отображение, которое каждому набору значений переменных ставит в соответствие объект сорта π , т.е. элемент множества D_π . Другими словами, терм получает значение, если известны значения всех его параметров. Значение терма легко определить индукцией по построению.

Обычно придание переменным терма некоторых значений рассматривается как некоторый формальный акт подстановки в терм вместо свободных переменных других объектов. Вообще говоря, использовать в выражении конкретные объекты не очень удобно. Во-первых, это выводит за пределы рассматриваемого языка. Во-вторых, подстановка в выражение „известно чего“ может привести к неоднозначностям или другим казусам, поскольку так называемые „реальные“ объекты вполне могут оказаться словами другого языка.

Указанный формальный подход можно реализовать следующим образом. Введем в обиход дополнительный набор констант, каждая из которых соответствует реальному объекту. Более точно, выберем множество $\underline{C}$ символов, равномощное $D = \bigcup_{\pi \in S} D_\pi$ и биективное отображение $\hat{C}: D \rightarrow \underline{C}$, устанавливающее взаимно однозначное соответствие между объектами $a \in D_\pi$ и символами $\underline{a}$ того же сорта. В формулах будем использовать не сами объекты, а сопоставленные им символы. В результате получим некоторое расширение $\Omega(D)$ исходного языка Ω .

Теперь мы можем в рамках расширенного языка выполнить формальную подстановку вида

$$\theta = \left(\begin{array}{ccc} x_1 & x_2 & \dots x_n \\ \underline{a}_1 & \underline{a}_2 & \dots \underline{a}_n \end{array} \right).$$

Это константная подстановка, называемая **оценкой языка** Ω . Если θ содержит все переменные терма T , то терм $T\theta$ оказывается замкнутым выражением (без параметров) и будет иметь конкретное значение. В этом случае θ называется **оценкой для выражения** T , а само выражение $T\theta$ называется **оцененным**. В общем оцененное выражение — это любое замкнутое выражение языка $\Omega(D)$.

Все сказанное переносится и на формулы языка Ω . Значением оцененной формулы является ложь или истина. При этом говорят, что оцененная формула истинна (ложна) в модели M . Подчеркнем, что сами по себе формулы не имеют какого-либо значения. Истинность возникает, когда конкретную интерпретацию получают все функциональные и предикатные символы, а формула не содержит параметров (или все они заменены константами в результате оценки).

Пример 3.1. В качестве примера логико-математического языка рассмотрим язык элементарной арифметики. Этот язык имеет лишь один сорт переменных, которые мы будем обозначать $x, y, z, \dots$. Единственную константу обозначим 0. Три функциональных символа f, g, h , из которых первый одноместный, два остальных двуместные. Единственный двуместный предикатный символ обозначим p .

Для функциональных и предикатного символов введем альтернативные бинарные нотации:

$f(x) \equiv x^+$ (переход к следующему натуральному числу);

$g(x, y) \equiv (x + y)$ (операция сложения);

$h(x, y) \equiv (x \cdot y)$ (операция умножения);

$p(x, y) \equiv (x = y)$ (формальное равенство).

Для введенного языка рассмотрим две модели. Первую обозначим ω . Носителем этой модели является множество $\mathbb{N}_0$ целых неотрицательных чисел. Символу 0 поставим в соответствие число нуль. Функциональным символам припишем соответствующие операции над целыми числами. Наконец, равенство $(x = y)$ будет истинным тогда и только тогда, когда два числа x и y совпадают.

Рассмотрим терм $(x + y) \cdot z + x^+$. Этому терму мы можем давать различные оценки. Например, при оценке $\begin{pmatrix} x & y & z \\ \underline{1} & \underline{2} & \underline{3} \end{pmatrix}$ получим терм $(\underline{1} + \underline{2}) \cdot \underline{3} + \underline{1}^+$, который в ω имеет значение 20. Формула $\exists y (x + y = z)$ при оценке $\begin{pmatrix} x & z \\ \underline{3} & \underline{5} \end{pmatrix}$ дает истинную оцененную формулу $\exists y (\underline{3} + y = \underline{5})$. В то же время при оценке $\begin{pmatrix} x & z \\ \underline{5} & \underline{3} \end{pmatrix}$ получим ложную формулу $\exists y (\underline{5} + y = \underline{3})$.

Носителем второй модели, которую обозначим R , является множество $\mathbb{R}$ действительных чисел. Функциональным символам опять сопоставим сложение, умножение и добавление единицы. Формула $x = y$ выражает совпадение действительных чисел.

Вообще говоря, в качестве интерпретации рассматриваемого языка можно выбрать любое множество D , два любых отображения типа $f, g: D^2 \rightarrow D$, отображение типа $h: D \rightarrow D$ и отображение $p: D \rightarrow \mathbb{B}$. Все возможно. Но при этом меняется смысл термов и формул. Например, в модели ω истинна формула $x + y = y + x$, отражающая арифметический закон. Она сохраняется в модели R , но теряет силу в других моделях.

3.5. Логические законы

Формулы общезначимые, выполнимые, опровержимые, тождественно ложные. Пропозициональные тавтологии. Тавтологии с кванторами. Пример: $\neg \exists x \neg X \rightarrow \forall x X$ — тавтология; $\forall x \exists y p(x, y) \rightarrow \exists y \forall x p(x, y)$ — нет. Основные логические законы: а) де Моргана (пронесение через отрицание); б) пронесение кванторов через бинарные связки (8 законов); в) обобщенное пронесение кванторов через дизъюнкцию и конъюнкцию (4 закона); г) добавление кванторов или изменение порядка (5 законов); д) законы конгруэнтности.

Среди формул языка Ω есть такие, которые остаются истинными при любой интерпретации и при любой оценке. Такие формулы называют **тавтологиями** или **общезначимыми формулами**. То, что X — тавтология, будем записывать так: $\models X$. Например, формула $p \rightarrow (q \rightarrow p)$ истинна в любой модели, поскольку вообще не имеет параметров и в любой модели автоматически становится оцененной, тождественно истинной формулой. В то же время есть формулы, которые являются тождественно истинными для данной модели, но не являются тождественно истинными в другой модели. Например, формула $x \cdot y = y \cdot x$ в модели ω тождественно истинна. Однако, если в качестве модели взять множество квадратных матриц (операции x^+ можно поставить в соответствие добавление единичной матрицы) с естественной интерпретацией двуместных функциональных символов, формула $x \cdot y = y \cdot x$ уже не будет тождественно истинной. Если формула X является истинной в данной модели M при любой оценке (т.е. тождественно истинна в M), но не является истинной в некоторой другой модели при некоторой оценке, то эта формула выражает **закон в модели M** , истинность X в модели M будем обозначать $M \models X$.

Поскольку нульарные предикатные символы не имеют параметров, их значение не меняется в рамках заданной модели. Поэтому множество всех нульарных предикатных символов в сочетании с логическими связками образует подмножество рассматриваемого языка, идентичное языку алгебры высказываний. Следовательно, любой тавтологии из алгебры высказываний соответствует тавтология языка Ω . Например, $\models \neg A \vee B \rightarrow \neg(A \wedge \neg B)$, так как это калька с тавтологии из алгебры высказываний. Такие тавтологии будем называть **пропозициональными**. В них нет ни предметных переменных, ни кванторов.

Кроме пропозициональных тавтологий в языке Ω есть и другие тавтологии, порожденные кванторами. Например $\neg\exists x\neg X \rightarrow \forall x X$ — тавтология. Чтобы это показать, выберем произвольную модель M и в ней произвольную оценку θ . Поскольку переменная x не входит свободно в формулу $\neg\exists x\neg X \rightarrow \forall x X$, можно считать, что x не входит и в верхнюю строку θ . Поэтому $(\neg\exists x\neg X \rightarrow \forall x X)\theta = \neg\exists x\neg X\theta \rightarrow \forall x X\theta$. Таким образом, необходимо установить истинность $M \models \neg\exists x\neg(X\theta) \rightarrow \forall x(X\theta)$. Для доказательства этого предположим, что левая часть импликации истинна, т.е. $M \models \neg\exists x\neg(X\theta)$. Надо доказать, что тогда $M \models \forall x(X\theta)$, т.е. для всякого объекта $a \in D_\pi$ верно $M \models (X\theta)_a^x$. Предполагая противное, заключаем, что существует такой объект $a \in D_\pi$, что формула $(X\theta)_a^x$ в M ложна. Но тогда $M \models \neg(X\theta)_a^x$. Следовательно $M \models \exists x\neg(X\theta)$. Но это противоречит предположению $M \models \neg\exists x\neg(X\theta)$.

Пример 3.2. Покажем, что формула $\forall x \exists y p(x, y) \rightarrow \exists y \forall x p(x, y)$ не является логическим законом. Для этого необходимо придумать такую модель M , что $M \models \forall x \exists y p(x, y)$, но формула $\exists y \forall x p(x, y)$ в модели M не является истинной (отметим, что рассматриваемые формулы замкнуты, а потому в любой модели автоматически являются оцененными).

В качестве модели выберем ω — модель элементарной арифметики. Предикатному символу $p(x, y)$ поставим в соответствие отношение $x < y$ на множестве натуральных чисел. Тогда формула $\forall x \exists y p(x, y)$ будет означать, что в множестве натуральных чисел с отношением порядка $x < y$ нет максимального элемента, а формула $\exists y \forall x p(x, y)$ будет иметь противоположный смысл, что в множестве натуральных чисел есть наибольший элемент. Ясно, что первая формула истинна, а вторая нет.

Две формулы X и Y называются **логически эквивалентными**, $X \equiv Y$, если формула $X \sim Y$ есть логический закон (напомним, что запись $X \sim Y$ — это сокращение формулы $(X \rightarrow Y) \wedge (Y \rightarrow X)$). Можно сказать иначе: две формулы логически эквивалентны, если в любой модели и при любой оценке они обе или истинны, или ложны.

Теорема 3.2. Если $X \equiv Y$, то $\forall x X \equiv \forall x Y$ и $\exists x X \equiv \exists x Y$.

► Пусть $x, x_1, x_2, \dots, x_n$ — полный набор всех параметров формул X и Y , в котором в качестве первой выбрана переменная x . При произвольно выбранной интерпретации языка и при указанном порядке переменных формулы X и Y порождают булевы функции $f_X(x, x_1, x_2, \dots, x_n)$ и $f_Y(x, x_1, x_2, \dots, x_n)$. Эквивалентность формул означает, что эти функции совпадают. Но тогда

$$\min_x f_X(x, x_1, \dots, x_n) = \min_x f_Y(x, x_1, \dots, x_n) \quad \text{и} \quad \max_x f_X(x, x_1, \dots, x_n) = \max_x f_Y(x, x_1, \dots, x_n).$$

Вычисление минимума или максимума функций f_X, f_Y равносильно навешиванию квантора $\forall$ или $\exists$ с переменной x на формулы X, Y . Это указывает на то, что пары формул $\forall x X$ и $\forall x Y$, $\exists x X$ и $\exists x Y$ эквивалентны в выбранной модели. Так как модель выбиралась произвольно, заключаем, что эти пары формул логически эквивалентны. ►

Приведем некоторые логические законы. Остановимся лишь на тех, которые выходят за рамки пропозициональных.

Законы де Моргана:

1. $\neg(\forall x X) \sim \exists x (\neg X)$.
2. $\neg(\exists x X) \sim \forall x (\neg X)$.

Доказательства общезначимости этих формул аналогичны доказательству общезначимости формулы $\neg\exists x\neg X \rightarrow \forall x X$, рассмотренной выше.

Одностороннее пренесение кванторов (здесь X не содержит свободно переменной x):

3. $X \wedge \forall x Y(x) \sim \forall x (X \wedge Y(x))$.
4. $X \vee \forall x Y(x) \sim \forall x (X \vee Y(x))$.
5. $X \wedge \exists x Y(x) \sim \exists x (X \wedge Y(x))$.
6. $X \vee \exists x Y(x) \sim \exists x (X \vee Y(x))$.
7. $X \rightarrow \forall x Y(x) \sim \forall x (X \rightarrow Y(x))$.
8. $X \rightarrow \exists x Y(x) \sim \exists x (X \rightarrow Y(x))$.
9. $\forall x Y(x) \rightarrow X \sim \exists x (Y(x) \rightarrow X)$.
10. $\exists x Y(x) \rightarrow X \sim \forall x (Y(x) \rightarrow X)$.

Рассмотрим, например, формулу 3. Выберем произвольную модель и в ней произвольную оценку θ . Поскольку x в формулу не входит свободно, считаем, что и в θ переменная x не выходит. Получим оцененную формулу $X\theta \wedge \forall x Y\theta \sim \forall x (X\theta \wedge Y\theta)$, которая является истинной в том и только в том случае, когда обе части эквиваленции имеют одно и то же истинностное значение. Если левая часть импликации истинна, то формулы $X\theta$ и $\forall x Y\theta$ обе истинны. Значит, при любом a формула $(Y\theta)_a^x$ является истинной. Следовательно, истинна формула $X\theta \wedge (Y\theta)_a^x$. Отсюда вытекает, что формула $\forall x (X\theta \wedge (Y\theta)_a^x)$ в правой части эквиваленции истинна. Нетрудно увидеть, что приведенные рассуждения можно провести в обратном порядке и тем самым показать, что из истинности правой части эквиваленции вытекает истинность левой части.

Отметим, что доказанная формула 3 имеет естественную интерпретацию при выборе конкретной модели:

$$\min \{f_X, \min_x f_Y\} = \min_x \min \{f_X, f_Y\}.$$

Это верно при условии, что функция f_X не зависит от переменной x .

Дополнительные законы пренесения кванторов:

11. $\forall x X(x) \wedge \forall x Y(x) \sim \forall x (X(x) \wedge Y(x))$. 13. $\exists x (X(x) \wedge Y(x)) \rightarrow \exists x X(x) \wedge \exists x Y(x)$.
 12. $\exists x X(x) \vee \exists x Y(x) \sim \exists x (X(x) \vee Y(x))$. 14. $\forall x X(x) \vee \forall x Y(x) \rightarrow \forall x (X(x) \vee Y(x))$.

Рассмотрим, например, формулу 11. Выбрав произвольную модель и в ней произвольную оценку θ (считаем, что она не содержит x), получим оцененную формулу $\forall x X\theta \wedge \forall x Y\theta \sim \forall x (X\theta \wedge Y\theta)$. Истинность левой части означает, что истинны формулы $\forall x X\theta$ и $\forall x Y\theta$. Следовательно, для любой константы a истинны $(X\theta)_a^x$ и $(Y\theta)_a^x$, откуда заключаем, что истинна формула $(X\theta)_a^x \wedge (Y\theta)_a^x$. Поэтому истинна $\forall x (X\theta \wedge Y\theta)$. Наоборот, если формула $\forall x (X\theta \wedge Y\theta)$ истинна, то для любой константы a истинна формула $(X\theta)_a^x \wedge (Y\theta)_a^x$. Делаем вывод, что для любой a истинна формула $(X\theta)_a^x$ и для любой a истинна формула $(Y\theta)_a^x$. Поэтому истинны формулы $\forall x X\theta$ и $\forall x Y\theta$, а значит, и формула $\forall x X\theta \wedge \forall x Y\theta$.

Логические законы, связанные добавлением кванторов или изменением их порядка:

15. $\forall x X \rightarrow X$. 18. $\forall x \forall y X \sim \forall y \forall x X$.
 16. $X \rightarrow \exists x X$. 19. $\exists x \forall y X \rightarrow \forall y \exists x X$.
 17. $\exists x \exists y X \sim \exists y \exists x X$.

Формулы 15 и 16 можно интерпретировать как утверждение, что для функции $u(x)$ имеет место неравенство

$$\min_x u(x) \leq u(x) \leq \max_x u(x). \quad (3.1)$$

Формулы 17 и 18 ассоциируются с правилом перестановки максимумов и минимумов по разным переменным, например:

$$\max_x \max_y u(x, y) = \max_y \max_x u(x, y),$$

а формула 19 — отражение неравенства

$$\max_x \min_y u(x, y) \leq \min_y \max_x u(x, y).$$

И в самом деле, $u(x, y) \leq \max_x u(x, y)$. Следовательно, $\min_y u(x, y) \leq \min_y \max_x u(x, y)$. В результате справа — константа, а слева — функция переменного x , т.е. $\min_y \max_x u(x, y)$ есть верхняя грань функции $\min_y u(x, y)$. Значит, имеет место неравенство (3.1).

Логические законы, связанные с конгруэнтностью (формула X не имеет свободных вхождений переменной y):

20. $\forall x X \sim \forall y X_y^x$. 21. $\exists x X \sim \exists y X_y^x$.

Эти логические законы указывают на то, что конгруэнтные формулы логически эквивалентны. Впрочем, эквивалентность конгруэнтных формул лучше доказывать непосредственно, опираясь на индуктивное определение конгруэнтности. Иначе придется доказывать, что конгруэнтная формула получается последовательным переименованием, что интуитивно ясно, но требует формальных рассуждений.

Что касается сформулированных логических законов, то можно рассуждать так (на пример квантора всеобщности). Выбираем произвольную модель и в ней произвольную оценку θ . При этом, поскольку ни x , ни y в формулу $\forall x X$ не входят свободно, можно считать, что и в оценку θ эти переменные не входят. Если переменная x не входит в X свободно, то наличие квантора не играет роли: формула $X\theta$ оказывается оцененной. Если же x входит в X свободно, то формула $X\theta$ будет содержать единственный параметр x . Истинность $\forall x(X\theta)$ означает, что для любого объекта $\underline{a}$ формула $(X\theta)_{\underline{a}}^x$ оказывается истинной. Но если мы предварительно выполним подстановку $(X\theta)_{\underline{y}}^x$, а затем подстановку $\left(\frac{y}{\underline{a}}\right)$, то получим ту же формулу $(X\theta)_{\underline{a}}^x$. Она окажется истинной. Аналогичны рассуждения при ложности формулы $\forall x(X\theta)$. Тем самым доказано, что в выбранной модели формулы $\forall x X$ и $\forall y X_{\underline{y}}^x$ при любой оценке одновременно истинны или ложны. Значит, они эквивалентны, а формула $\forall x X \sim \forall y X_{\underline{y}}^x$ есть логический закон.

3.6. Замены

Понятие формального предиката. Замена элементарной формулы формальным предикатом (индуктивное определение). Теорема о замене элементарных подформул. Правило замены эквивалентным. Пример: эквивалентность $r(x) \vee \forall z (\neg \forall x q(x, z))$ и $r(x) \vee \forall z (\exists x \neg q(x, z))$. Принцип двойственности. Нильпотентность. Эквивалентность двойственных формул.

Речь идет о получении аналога теоремы 1.2. У нас нет самостоятельных пропозициональных переменных. Их роль играют предикатные символы арности нуль. Разумеется, теорема 1.2 остается верной при замене предикатного символа арности нуль какой-либо формулой. Но как быть, если нужно заменить предикатный символ, имеющий параметры? Таким образом, речь идет об обобщении теоремы 1.2 на случай замены в формуле произвольной элементарной подформулы. Механизм замены должен сохранять эквивалентность формул.

При замене какого-либо вхождения элементарной подформулы $p(t_1, \dots, t_n)$ (аргументы в этой подформуле — какие-либо термы) некоторой формулой X следует установить соответствие между переменными, входящими в элементарную подформулу, и переменными, входящими в X . Такое соответствие можно организовать, вводя понятие **формального предиката**. Под формальным предикатом мы будем понимать слово вида $x_1 x_2 \dots x_n X$. Такое слово означает лишь, что задана какая-то формула X и упорядоченный набор переменных $x_1, x_2, \dots, x_n$ — больше ничего. Эти переменные и все их вхождения в X в рамках формального предиката будем считать связанными. Кортеж сортов $(\pi_1, \pi_2, \dots, \pi_n)$ „объявленных“ переменных $x_1, x_2, \dots, x_n$ будем называть типом формального предиката.

Чтобы в формуле Y заменить элементарную подформулу $p(t_1, \dots, t_n)$, необходимо построить формальный предикат $U = x_1 x_2 \dots x_n X$ того же типа, что и предикатный символ p , затем в формуле X выполнить подстановку $X_{t_1, \dots, t_n}^{x_1, \dots, x_n}$ и результат подстановки подставить в формулу Y взамен подформулы $p(t_1, \dots, t_n)$.

Подобный подход будет более понятным, если учесть, что в любой модели формула задает предикат, роль аргументов которого играют переменные формулы. Задавая формальный предикат, мы выбираем некоторую совокупность переменных (записываем их перед формулой) и упорядочиваем. Эти отобранные переменные фиксируются, остальные остаются свободными. При подстановке вместо элементарной формулы мы должны на место „объявленных аргументов“ предиката подставить аргументы элементарной формулы и в таком виде подставить формулу вместо элементарной. Например, пусть элементарная формула имеет вид $p(t_1, t_2, \dots, t_n)$. Формальный предикат $x_1 x_2 \dots x_n X$ следует в любой модели интерпретировать как предикат $\tilde{p}(x_1, \dots, x_n, z_1, \dots, z_m)$, в котором первые n аргументов по типам соответствуют аргументам предиката p . В результате мы меняем в исходной формуле элементарную подформулу $p(t_1, t_2, \dots, t_n)$ на подформулу $\tilde{p}(t_1, \dots, t_n, z_1, \dots, z_m)$. Отобранные элементы формального предиката играют роль подстановочных мест и заменяются аргументами элементарной подформулы. Оставшиеся переменные формулы X без изменений переходят в изменяемую формулу.

Указанный алгоритм подстановки можно описать с помощью индукции по построению формулы. Пусть задан формальный предикат $U = x_1 x_2 \dots x_n X$ и Y обозначает исходную формулу. Результат $Y(p \parallel U)$ подстановки в Y формального предиката U типа $(\pi_1, \pi_2, \dots, \pi_n)$ вместо элементарной подформулы $p(t_1, \dots, t_n)$ с предикатным символом p того же типа определим следующим образом.

1. Если Y — атомарная формула $q(r_1, \dots, r_m)$, то при q , отличном от p полагаем $Y(p \parallel U) = Y$, а если $Y = p(t_1, \dots, t_n)$, то полагаем $Y(p \parallel U) = X\theta$, где $\theta = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ t_1 & t_2 & \dots & t_n \end{pmatrix}$.

2. Если $Y = V\nabla W$, где ∇ — одна из трех двуместных логических связок, то $Y(p \parallel U) = V(p \parallel U)\nabla W(p \parallel U)$. Если $Y = \neg V$, то $Y(p \parallel U) = \neg V(p \parallel U)$.

3. Если $Y = \nabla z V$, где ∇ — один из кванторов, то возможны два случая. В первом случае, когда U не содержит свободных вхождений z или V не содержит вхождений предикатного символа p , полагаем $Y(p \parallel U) = \nabla z V(p \parallel U)$. Во втором случае, когда z имеет свободные вхождения в U , а V имеет вхождения p , выбираем новую переменную u (т.е. не встречающуюся ни в Y , ни в U) и полагаем $Y(p \parallel U) = \nabla u (V_u^z)(p \parallel U)$.

Обратим внимание на то, что в соответствии с описанным индуктивным правилом при замене меняются все вхождения данного предикатного символа. Последний пункт регулирует вхождение свободных переменных формального предиката. Снова действует правило: никакое свободное вхождение переменной после преобразования не должно оказаться связанным. Если такое получается, соответствующую кванторную переменную надо изменить.

Теорема 3.3. Если $X \equiv Y$, то и $X(p \parallel x_1 \dots x_k Z) \equiv Y(p \parallel x_1 \dots x_k Z)$. Если $Y \equiv Z$, то и $X(p \parallel x_1 \dots x_k Y) \equiv X(p \parallel x_1 \dots x_k Z)$.

◀ Первое утверждение сводится к следующему: если U — тавтология, то и $U(p \parallel x_1 \dots x_k Z)$ — тавтология (достаточно в качестве U взять формулу $X \sim Y$). То, что U — тавтология, означает, что при любой интерпретации и любой оценке формула U истинна. Интерпретация, в частности, обеспечивает конкретное содержание предикатного символа p , а формальный предикат $x_1 x_2 \dots x_k Z$ превращается в некоторый предикат $q(x_1, \dots, x_k, y_1, \dots, y_l)$. При замене происходит замена одного предиката другим, при этом появляются свободные переменные $y_1, \dots, y_l$, которые мы можем зафиксировать с помощью оценки. В результате дело фактически сводится к замене предикатной буквы p на предикатную букву q . Такую замену можно интерпретировать как изменение интерпретации буквы p . При изменении интерпретации формула остается истинной. Значит, и после замены формула остается истинной.

Доказательство второго утверждения аналогично. Замена предикатного символа p одним из формальных предикатов можно интерпретировать как выбор интерпретации символа p . Так как формулы Y и Z эквивалентны, то и первая, и вторая замены дают одну и ту же интерпретацию символу p . Следовательно и истинностное значение двух формул будет одинаковым. ▶

Следствие 3.1. Если $Y \equiv Z$, то после замены в формуле X одного из вхождений подформулы Y формулой Z получим формулу X' , эквивалентную X .

◀ Пусть $x_1, x_2, \dots, x_n$ — полный список переменных, входящих в формулы Y и Z . Рассмотрим формулу Γ , полученную заменой подформулы X элементарной формулой $p(x_1, x_2, \dots, x_n)$. Тогда $X = \Gamma(p \parallel x_1 \dots x_n Y)$ и $X' = \Gamma(p \parallel x_1 \dots x_n Z)$. Согласно теореме эти формулы эквивалентны. ▶

Утверждение следствия известно как **правило замены эквивалентным**.

Пример 3.3. Покажем, что формулы $r(x) \vee \forall z (\neg \forall x q(x, z))$ и $r(x) \vee \forall z (\exists x \neg q(x, z))$ эквивалентны.

Обе формулы можно рассматривать как две подстановки в формулу $X = r(x) \vee \forall z p(z)$ двух эквивалентных формул $V = \neg \forall x q(x, z)$ и $W = \exists x \neg q(x, z)$. Сформируем два формальных предиката zV и zW . В результате получим $X(p \parallel zV) = r(x) \vee \forall z (\neg \forall x q(x, z))$ и $X(p \parallel zW) = r(x) \vee \forall z (\exists x \neg q(x, z))$. В соответствии с последним логическим законом эти формулы эквивалентны.

Отметим, что было бы неверно в качестве шаблона использовать, например, формулу $X_1 = r(x) \vee \forall z p$, опуская в заменяемом символе связанную переменную z . В этом случае в соответствии с индуктивным правилом мы получили бы формулы $X_1(p \parallel zV) = r(x) \vee \forall z_1 (\neg \forall x q(x, z))$ и $X_1(p \parallel zW) = r(x) \vee \forall z_1 (\exists x \neg q(x, z))$, которые тоже эквивалентны, но отличаются от того, что мы хотели бы получить.

На формулы алгебры предикатов, не содержащие импликации, распространяется принцип двойственности. Формулу X^* , двойственную данной формуле X определим индуктивно:

- 1) если X — элементарная формула, то $X^* = X$;
- 2) если $X = \neg U$, то $X^* = \neg U^*$;
- 3) если $X = U \wedge V$, то $X^* = U^* \vee V^*$;
- 4) если $X = U \vee V$, то $X^* = U^* \wedge V^*$;
- 5) если $X = \forall x U$, то $X^* = \exists x U^*$;
- 6) если $X = \exists x U$, то $X^* = \forall x U^*$.

Непосредственно из определения следует, что преобразование двойственности нильпотентно: $X^{**} = X$. Следующее утверждение связывает преобразование двойственности с отношением эквивалентности. Кроме того, верно следующее утверждение. Назовем внутренним отрицанием $\check{X}$ формулы X ее преобразование, при котором каждая элементарная подформула $p(t_1, t_2, \dots, t_n)$ заменяется формальным предикатом $x_1 x_2 \dots x_n \neg p(x_1, x_2, \dots, x_n)$.

Теорема 3.4. Формула X^* , двойственная формуле X , эквивалентна формуле $\neg \check{X}$.

◀ Доказательство проводится индукцией по построению формулы. Для элементарных формул внутреннее отрицание совпадает с обычным отрицанием $\neg$, так что $\neg \check{X} = X$, но в то же время по определению $X^* = X$.

Если $X = U \wedge V$ и уже доказано, что $U^* \equiv \neg \check{U}$, $V^* \equiv \neg \check{V}$, то согласно определению двойственной формулы и правилу замены эквивалентным

$$X^* = U^* \vee V^* \equiv \neg \check{U} \vee \neg \check{V} \equiv \neg(\check{U} \wedge \check{V}) = \neg \check{X}.$$

Аналогично утверждение доказывается для дизъюнкции и отрицания.

Если $X = \forall x U$ и уже доказано, что $U^* \equiv \neg \check{U}$, то

$$X^* = \exists x U^* \equiv \exists x \neg \check{U} = \neg \forall x \check{U} = \neg \check{X}.$$

Аналогичны рассуждения для кванторной приставки $\exists x$. ▶

Теорема 3.5. Эквивалентность $X \equiv Y$ равносильна эквивалентности $X^* \equiv Y^*$.

◀ Доказательство в силу нильпотентности достаточно провести в одну сторону. При этом достаточно заметить, что если $X \equiv Y$, то $\check{X} \equiv \check{Y}$ и $\neg X \equiv \neg Y$. Поэтому в силу предыдущей теоремы

$$X^* \equiv \neg \check{X} \equiv \neg \check{Y} \equiv Y^*. \quad \blacktriangleright$$

3.7. Упрощение формул

Приведение бескванторной формулы к ДНФ и КНФ. Предваренная нормальная форма. Теорема о приведении. Пример: $X = \forall x \neg \exists y p(x, y) \rightarrow \forall x (q(x) \rightarrow \neg \exists y p(x, y))$. Замечание о порядке вынесения кванторов.

На основании логических законов можно преобразовывать формулы логико-математического языка, упрощая или добиваясь получения формул определенной структуры. Например, если в формуле отсутствуют кванторы (бескванторная формула), то ее можно привести к ДНФ или КНФ. Такое преобразование всегда можно проводить в рамках языка логики высказываний, рассматривая любую элементарную подформулу как пропозициональную переменную.

Теорема 3.6. Всякая бескванторная формула эквивалентна некоторой ДНФ и некоторой КНФ.

Разумеется, подобные преобразования возможны и для формул, содержащих кванторы. Наличие кванторов лишь усложняет сам процесс, но не отменяет его.

Кванторы не мешают преобразованиям, если они в формуле собраны вместе. Формулу вида $\nabla x_1 \nabla x_2 \dots \nabla x_n X$, где X — бескванторная формула, называют **предваренной** или **пренексной**. В частности, любая бескванторная формула является предваренной. Предваренной (пренексной) формой данной формулы X называют любую предваренную формулу Y , логически эквивалентную X . Часть формулы Y без кванторных приставок называют **матрицей предваренной формы**.

Теорема 3.7 (о предваренной форме). Любая форма имеет предваренную форму.

◀ Для доказательства нужно исходную формулу X заменить формулой X' со свойством чистоты переменных, а затем, применяя логические законы одностороннего пронесения кванторов, „вытащить“ все кванторы за пределы формулы.

Аккуратно доказательство следует проводить методом индукции по построению формулы. В самом деле, утверждение очевидно для элементарных формул (они все бескванторные). Пусть $X = V \circ W$, где $\circ$ — одна из двуместных логических связей, и формулы V и W имеют предваренные формы: $V \equiv \nabla v_1 \dots \nabla v_m V'$, $W \equiv \nabla w_1 \dots \nabla w_k W'$, где V' и W' бескванторные. Полагая, что X обладает свойством чистоты переменных, заключаем, что все переменные v_i и w_j различны, V' не содержит переменных w_j , а W' — переменных v_i . Тогда с помощью законов одностороннего пронесения кванторов получаем

$$X \equiv (\nabla v_1 \dots \nabla v_m V') \circ (\nabla w_1 \dots \nabla w_k W') \equiv \nabla v_1 \dots \nabla v_m \nabla w_1 \dots \nabla w_k (V' \circ W').$$

Здесь мы воспользовались также правилом замены подформулы эквивалентной.

Если формула X имеет вид $X = \neg V$, рассуждения аналогичны. Наконец, в случае $X = \nabla z V$ достаточно заменить V предваренной формой, чтобы получить предваренную форму X . ►

Пример 3.4. Рассмотрим формулу $X = \forall x \neg \exists y p(x, y) \rightarrow \forall x (q(x) \rightarrow \neg \exists y p(x, y))$. Она не обладает свойством чистоты. Используя переименование с помощью двух новых переменных u и v , получим конгруэнтную (а потому эквивалентную) формулу

$$X' = \forall x \neg \exists y p(x, y) \rightarrow \forall u (q(u) \rightarrow \neg \exists v p(u, v)).$$

Применяя одностороннее пронесение кванторов, последовательно получаем:

$$\begin{aligned} & \forall x \neg \exists y p(x, y) \rightarrow \forall u (q(u) \rightarrow \neg \exists v p(u, v)) \equiv \\ & \equiv \forall x \forall y \neg p(x, y) \rightarrow \forall u (q(u) \rightarrow \forall v \neg p(u, v)) \equiv \\ & \equiv \forall x \forall y \neg p(x, y) \rightarrow \forall u \forall v (q(u) \rightarrow \neg p(u, v)) \equiv \\ & \equiv \exists x \exists y (\neg p(x, y) \rightarrow \forall u \forall v (q(u) \rightarrow \neg p(u, v))) \equiv \\ & \equiv \exists x \exists y \forall u \forall v (\neg p(x, y) \rightarrow (q(u) \rightarrow \neg p(u, v))). \end{aligned}$$

Замечание. Вынесение кванторов может проходить в разном порядке. Следовательно, и в предваренной форме кванторные приставки могут идти в разном порядке: предваренная форма определена неоднозначно. В матрице предваренной формы все связки идут в том же порядке и без изменений. Она получается из исходной формулы удалением всех кванторных приставок.

ОГЛАВЛЕНИЕ

1. Алгебра высказываний	1
1.1. Введение	1
1.2. Алгебра логики	3
1.3. Тавтологии и эквивалентность формул	5
1.4. Функции алгебры логики	9
2. Исчисление высказываний	14
2.1. Основные положения теории N	14
2.2. Правила естественного вывода	16
2.3. Глобальные свойства теории N	21
3. Алгебра предикатов	26
3.1. Предикаты и кванторы	26
3.2. Логико-математические языки	27
3.3. Переименования и подстановки	30
3.4. Семантика логико-математического языка	33
3.5. Логические законы	35
3.6. Замены	38
3.7. Упрощение формул	40
4. Исчисление предикатов	42
4.1. Построение теории P	42
4.2. Правила естественного вывода	43
4.3. Глобальные свойства теории P	45
5. Генценовские формальные системы	53
5.1. Исчисление GV	53
5.2. Исчисление GP	57
6. Примеры формальных теорий	60
6.1. Теория групп	60
6.2. Формальная арифметика	63
7. Метод резолюций	69
7.1. Скулемовские функции	69
7.2. Метод резолюций для исчисления высказываний	71
7.3. Метод резолюций для исчисления предикатов	74