Специальные главы математики

КОНСПЕКТ ЛЕКЦИЙ

Предисловие*

Материалы, помеченные значком ▽, — необязательны.

Предмет дискретной математики

Дискретная математика — часть математики, которая зародилась в глубокой древности. Как говорит само название, главной ее спецификой является дискретность, т.е. антипод непрерывности.

Дискретный подход к математическим проблемам преобладал до середины XIX века (арифметика — это дискретная математика, а известный немецкий математик Леопольд Кронекер сказал, что "бог создал целые положительные числа, а все остальное придумал человек"), когда успехи развития математического анализа придали непрерывному подходу первостепенное значение. Но со второй половины XX века дискретный подход опять потеснил непрерывный. Это случилось во многом из-за того, что абсолютное большинство современных ЭВМ построены на дискретных принципах.

В курсе будут рассмотрены следующие разделы дискретной математики: теория множеств, отношений и функций; введение в теорию графов; теория групп, полей и колец; теория функций алгебры логики; математическая логика; элементы теории алгоритмов.

Теория множеств

Алгебраическая система — это множество некоторых элементов и набор операций, применимых к этим элементам так, что результаты этих операций есть элементы, принадлежащие исходному множеству. (Понятие алгебраической системы является одним из основных в современной математике.)

Множество — это неупорядоченная совокупность различных элементов.

(П) Понятие множества очень часто используется в повседневной речи, где для него существует много синонимов: класс, толпа, стая, совокупность. В математике понятие множества принадлежит к числу самых фундаментальных. Примеры математических множеств: все натуральные числа — $\mathbb{N}$, все целые числа — $\mathbb{Z}$, все рациональные числа — $\mathbb{Q}$, все вещественные числа — $\mathbb{R}$, все комплексные числа — $\mathbb{C}$. Каждое из названных множеств бесконечно и, кроме того, образует алгебраическую систему относительно операций сложения и умножения.

Множества могут содержать конечное и бесконечное число элементов. В первом случае множества называют **конечными**, во втором —

---

* Для подготовки материалов использовались системы Plain TEX, AMSFonts, PICTEX и TreeTEX

бесконечными. Конечное множество можно задать простым перечислением всех его элементов.

(П) $\{1, 2, 7\}$ и $\{2, 7, 1\}$ задают одно и то же множество.

Если бесконечное множество задается формулой вычисления n-го элемента, то его называют **счетным**.

(П) Множество всех неотрицательных целых чисел можно описать как $\mathbb{Z}_+ = \{0, 1, \ldots (n-1), \ldots\}$.

Чтобы **определить** некоторое множество S, нужно объяснить, как отвечать на следующий вопрос: принадлежит ли данный объект a множеству S или нет? Вместо "a принадлежит S" можно сказать "a является элементом S" и записывать этот факт символически, $a \in S$. Если a не принадлежит S, то символически это записывается как $a \notin S$.

Подмножеством S некоторого множества T называется любое множество S, все элементы которого принадлежат T. Иными словами, из $a \in S$ следует, что $a \in T$. Это отношение между S и T символически записывается как $S \subset T$ (или $T \supset S$). Иногда используют несколько иное обозначение, $S \subseteq T$ (или $T \supseteq S$). Про это отношение говорят, что S содержится (включается) в T.

(У1) Отношение включения $\subset$ обладает рядом свойств: 1) $\forall S, S \subset S$ (рефлексивное свойство включения или рефлексивность); 2) из $S \subset T$ и $T \subset U$ следует, что $S \subset U$ (транзитивное свойство включения или транзитивность). [Очевидно]

Всякое множество однозначно определяется своими элементами. Иными словами, множества S и T **совпадают**, когда они обладают в точности одними и теми же элементами, т.е. множества S и T **равны**, когда элемент x принадлежит S в том и только в том случае, когда x принадлежит T. Символически равенство множеств S и T записывается как $S = T$.

(У2) $S = T \quad \Leftrightarrow \quad S \subset T$ и $T \subset S$. [Включение $S \subset T$ имеет место в том и только в том случае, когда $x \in S \Rightarrow x \in T$, и включение $T \subset S$ имеет место в том и только в том случае, когда $x \in T \Rightarrow x \in S$, что и доказывает утверждение].

Наиболее часто подмножество S некоторого множества U определяется как множество всех тех элементов $x \in U$, которые обладают некоторым определенным **свойством**. Свойство — это утверждение относительно x, обозначаемое обычно как $P(x)$. Итак, символически определение S можно записать как $S = \{x \in U \mid P(x)\}$ или $S = \{x \mid P(x)\}$. Последние формулы читаются так: "S есть множество всех элементов x множества U, для которых справедливо утверждение $P(x)$".

(П) Формулы $2\mathbb{Z} = \{x \in \mathbb{Z} \mid \exists y \in \mathbb{Z}\ x = 2y\}$ и $\mathbb{Z}_+ = \{x \in \mathbb{Z} \mid x \geqslant 0\}$ описывают множество всех четных и неотрицательных целых чисел

соответственно.

Множество называется **пустым**, если оно не содержит ни одного элемента. Пустое множество обозначается знаком $\varnothing$.

Множество всех подмножеств (частей) множества U обозначается через $P(U)$ и называется **множеством-степенью**. Множество-степень всегда содержит в качестве своих элементов само U и пустое множество.

(П) Множество всех частей $P(\{a,\, b,\, c\})$ множества $U = \{a,\, b,\, c\}$ содержит следующие подмножества: $\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}$, U.

(УЗ) Если U состоит из n элементов $a_1, a_2, \ldots, a_n$, то $P(U)$ состоит из 2^n подмножеств $S_1, S_2, \ldots, S_{2^n}$. [Этот подсчет основан на том, что для каждого элемента $a_i \in U$ $(1 \leqslant i \leqslant n)$ и каждого подмножества $S_k \subset U$ имеет место ровно одна из двух возможностей: $a_i \in S_k$ или $a_i \notin S_k$. Иными словами, любое множество S_k задается цепочкой длины n из нулей и единиц, где 0 в i-й позиции означает, что $a_i \notin S_k$, а 1 — $a_i \in S_k$. Очевидно, что всех таких цепочек будет ровно 2^n, а это и требовалось доказать.]

Подмножества U, не равные U и $\varnothing$, называются **собственными**.

Парадоксы теории множеств

Интуитивное понятие множества к расстройству всех математиков оказалось внутренне логически противоречивым. Это было продемонстрировано великим математиком и философом Бертраном Расселом в его знаменитом парадоксе (парадокс Рассела-Цермело).

Итак, бывают множества, содержащиеся в себе как элементы, например, множество всех множеств, и, очевидно бывают множества, таким свойством не обладающие, например, множество $\{1, 2\}$. Пусть теперь A — это множество таких множеств X, для которых X не есть элемент X, $A = \{X \mid X \notin X\}$. Пусть $A \in A$, тогда $A \notin A$. Аналогично, если $A \notin A$, то $A \in A$.

Другой парадокс часто приводится в форме задачи про деревенского брадобрея, который бреет в деревне всех, кто не бреется сам. Любой ответ на вопрос: “Брить ли брадобрею себя?” — приводит к противоречию.

Рассмотрим еще один известный парадокс (Ришара). Пусть множество X содержит все натуральные числа, кратчайшее определение которых состоит менее чем из 100 символов, например, кратчайшее определение числа 1222 (“одна тысяча двести двадцать два”) содержит не более 31 символа. Множество X — конечно, потому что существует лишь конечное число комбинаций длины, меньшей 100, из конечного числа символов. Значит существуют натуральные числа, не входящие в

X. Пусть r наименьшее такое число, оно описывается фразой "наименьшее натуральное число, которое не может быть определено, используя менее 100 символов", которая содержит менее 100 символов...

Попытки устранения подобных парадоксов привели к созданию в начале XX века аксиоматической теории множеств. Однако проблема полного избавления математики от парадоксов не решена и по сей день.

Рассмотренные парадоксы разрешаются вводом в понятие множества дополнительного ограничения: "Множество не должно содержать элементов, определяемых через него самого".

Операции с множествами

Бинарная операция на множестве S — это правило, которое ставит в соответствие каждой упорядоченной паре элементов из S третий элемент из S — значение этой операции на заданной паре.

(П) Сложение $(+)$ и вычитание $(-)$ в арифметике — это бинарные операции на множестве всех целых чисел, $\mathbb{Z}$.

Унарная операция на множестве S — это правило f, которое каждому элементу a из S ставит в соответствие элемент $f(a) \in S$. Унарная операция на S — это функция f с областью определения S и областью значений, включающейся в S.

(П) Унарный минус на множестве $\mathbb{Z}$, целых чисел — это типичная унарная операция. Не путать унарные и бинарные минусы!

На множестве всех подмножеств любого данного множества U определены **три** фундаментальные операции: две бинарные и одна унарная. Это операции **теоретико-множественного** пересечения, объединения и дополнения. Эти операции часто называют булевыми (по имени английского математика XIX века Джорджа Буля).

Пересечением двух множеств S и R называют множество, состоящее только из тех элементов, которые входят как в S, так и в R. Символически это записывается как $S \cap R = \{x \mid x \in S \text{ и } x \in R\}$.

Объединением двух множеств S и R называют множество, состоящее из тех элементов, которые входят либо в S, либо в R. Символически это записывается как $S \cup R = \{x \mid x \in S \text{ или } x \in R\}$.

Дополнением (абсолютным) множества S до множества U называют множество S', состоящее из тех элементов U, которые не входят в S. Символически это записывается как $S' = \{x \in U \mid x \notin S\}$.

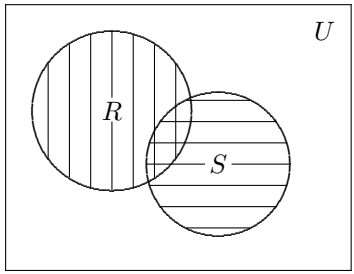
Множества S и T называются **дополнительными**, если $T = S'$ или, что то же самое, $S = T'$.

Относительным дополнением множества S до множества R называется множество тех элементов R, которые не принадлежат S. Символически это записывается так: $R \backslash S = \{x \in R \mid x \notin S\}$. Относительное дополнение $R \setminus S$ называют также разностью множеств R и S, $R - S$.

Очевидно, что $R \setminus S = R - S = R \cap S'$.

(П) Пусть $2\mathbb{Z} = \{0, \pm 2, \pm 4, \ldots\}$ — множество всех четных целых чисел. Тогда $2\mathbb{Z} \cap \mathbb{N} = \{2, 4, 6, \ldots\}$ — это множество всех положительных четных чисел, $2\mathbb{Z} \cup \mathbb{N}$ — это множество всех целых чисел, не являющихся одновременно четными и отрицательными.

Для обозначения операций с множествами часто используют **диаграммы Венна** (или Эйлера-Венна). На этих диаграммах U — это прямоугольник, а подмножества S и R — это круги. На рисунке пересечение S и R — это самая темная область, а вся заштрихованная область — это объединение S и R, а вся незаштрихованная область — $(S \cup R)'$.



Обозначим через $\#S$ **число элементов множества** S.

(У4) Если R и S конечные множества, не имеющие общих элементов (т. е. $S \cap R = \varnothing$), то число элементов объединения S и R равно сумме числа элементов S и R, $\#(S \cup R) = \#S + \#R$. [Очевидно]

(У5) Для любых конечных множеств R и S справедливо более общее равенство, $\#(S \cup R) = \#S + \#R - \#(S \cap R)$. [Действительно, $\#(S \cap R)$ — это число тех элементов, которые считаются дважды при последовательном пересчете элементов R и S.]

Когда $S \cap R = \varnothing$, говорят, что S и R **не пересекаются**.

(У6) Три операции $\cup$, $\cap$ и $'$ на множестве всех частей фиксированного множества U удовлетворяют ряду основных алгебраических законов:

1. $S \cap S = S$, $S \cup S = S$ (законы идемпотентности);
2. $S \cap R = R \cap S$, $S \cup R = R \cup S$ (законы коммутативности);
3. $(S \cap R) \cap T = S \cap (R \cap T)$, $(S \cup R) \cup T = S \cup (R \cup T)$ (законы ассоциативности);
4. $S \cap (S \cup R) = S \cup (S \cap R) = S$ (закон поглощения);
5. Из $R \subset T$ следует $R \cup (S \cap T) = (R \cup S) \cap T$ (модулярный закон);
6. $R \cap (S \cup T) = (R \cap S) \cup (R \cap T)$, $R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$ (законы дистрибутивности);
7. $R \cap \varnothing = \varnothing$, $R \cup \varnothing = R$, $R \cup U = U$, $R \cap U = R$ (универсальные границы);
8. $R \cap R' = \varnothing$, $R \cup R' = U$ (дополняемость);
9. $(S')' = S$ (инволютивный закон);
10. $(S \cap R)' = S' \cup R'$, $(S \cup R)' = S' \cap R'$ (законы де Моргана).

[Законы 1, 2, 4, 5, 7, 8 и 9 доказываются на диаграммах Венна, 6 и 10 в [13, стр. 9]]

(П) Подобно тому как целые числа образуют алгебраическую систему относительно двух бинарных операций сложения и умножения и одной унарной операции, минус; множество всех частей любого множества U, $P(U)$, образует алгебраическую систему относительно трех фундаментальных теоретико-множественных операций.

Представление множества U в виде попарно не пересекающихся непустых подмножеств называется **разбиением** U.

(П) S и S' образуют разбиение U.

Алгебраическая система, три операции которой удовлетворяют законам 1–10, называется **булевой алгеброй**.

(У7) Множество всех частей $P(U)$ любого множества U является булевой алгеброй, относительно трех фундаментальных теоретико-множественных операций. [Следует из определения булевой алгебры и свойств фундаментальных теоретико-множественных операций.]

(П) Пусть $U = \{a\}$ — одноэлементное множество, тогда $P(U) = \{\varnothing, U\}$. Булева алгебра задается таблицами операций:

$\cap$	$\varnothing$	U
$\varnothing$	$\varnothing$	$\varnothing$
U	$\varnothing$	U

$\cup$	$\varnothing$	U
$\varnothing$	$\varnothing$	U
U	U	U

$'$	
$\varnothing$	U
U	$\varnothing$

.

Если обозначить $\varnothing$ 0, а U — 1, то таблица операций примет следующий вид:

$\cap$	0	1
0	0	0
1	0	1

$\cup$	0	1
0	0	1
1	1	1

$'$	
0	1
1	0

.

Итак, установлено соответствие теоретико-множественных операций $\cup$, $\cap$ и $'$ логическим операциям OR, AND и NOT соответственно.

(П) Требуется составить булеву алгебру для $P(U)$, $U = \{a, b\}$. Ясно, что $P(U) = \{\varnothing, \{a\}, \{b\}, U\}$. Обозначим $\varnothing$ через 00, $\{a\}$ через 10, $\{b\}$ — 01 и U — 11. И далее, на основе полученных ранее соотношений, таблицы получаются автоматически.

Функции и отношения

Пусть S и T — множества. **Функцией** f из области S в область T называется правило, которое сопоставляет каждому элементу $s \in S$ единственный элемент из T, называемый значением f в s и обозначаемый $f(s)$. Функцию также называют **отображением** S в T и обозначают $f\colon S \to T$. Величина s называется **аргументом** функции, S — **областью определения** функции, а T — **областью, содержащей значения** функции.

Образом $Im\, f$ отображения $f\colon S \to T$ называется множество всех значений $f(s)$, которые оно принимает при всевозможных $s \in S$. Оче-

видно, что $Im\, f \subset T$. Образ f — это ее **множество значений**.

Функция называется **тождественной**, если она переводит каждый элемент области определения S в себя. Символически тождественную функцию будем обозначать 1_S. Тождественные функции для разных множеств различны.

Композицией $g \circ f$ двух функций f и g называется функция, полученная в результате их применения в порядке обратном написанному (сначала применяется f, а затем g). Или формально, пусть $f\colon S \to T$ и $g\colon T \to U$, тогда $g \circ f\colon S \to U$ определяется правилом $(g \circ f)(s) = g(f(s))$ для всех $s \in S$. Символ композиции для краткости часто опускают, т.е. допустима запись $(fg)h$ вместо $(f \circ g) \circ h$.

(У8) Композиция функций подчиняется ассоциативному закону, т.е. $(f \circ g) \circ h = f \circ (g \circ h)$. [$((f \circ g) \circ h)(x) = (f \circ g)(h(x)) = f(g(h(x)))$, $(f \circ (g \circ h))(x) = f((g \circ h)(x)) = f(g(h(x)))$.]

(У9) Тождественные функции 1_S и 1_T в композиции с любой функцией $f\colon S \to T$ не меняют ее, $f \circ 1_S = 1_T \circ f = f$. [Очевидно]

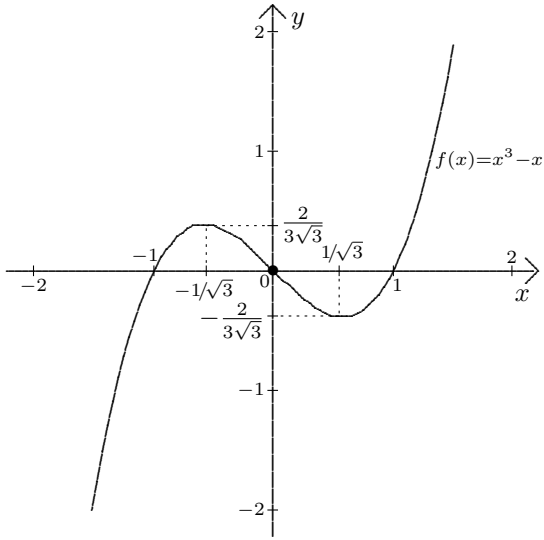
Функция $f\colon S \to T$ называется **инъективной** или **инъекцией**, если из $s_1 \neq s_2$ для $s_1 \in S$ и $s_2 \in S$ следует, что $f(s_1) \neq f(s_2)$. Инъекция переводит различные элементы своей области определения в различные элементы своего образа.

Функция $f\colon S \to T$ называется **сюръективной** или **сюръекцией**, если $Im\, f = T$, т.е., если $\forall t \in T\ \exists s \in S$ такой, что $f(s) = t$.

Функция $f\colon S \to T$ называется **биективной** или **биекцией**, если она одновременно и сюръекция и инъекция. Биективные функции называют также взаимно-однозначными.

(П) Среди функций из $\mathbb{Z}$ в $\mathbb{Z}$ отображение $f(n) = -n$ биективно; отображение $f(n) = 2n$ инъективно, но не сюръективно; а отображение $f(n) = n^2$ не является ни сюръекцией, ни инъекцией.

(П) Функция $f(x) = x^3 - x$, определенная на множестве $\mathbb{R}$ — это не инъекция, но сюръекция. [Сюръективность очевидна из графика этой функции.]



Пусть $X_n = \{1, 2, \ldots, n\}$. К числу основных функций на нем принадлежит **циклическая перестановка**, $\varphi_n\colon X_n \to X_n$. Она сопоставляет любому числу $k < n$, следующее $(k+1)$, а n переводит в 1. Это биекция.

(П) Пусть $X_3 = \{1, 2, 3\}$, тогда $\varphi_3(1) = 2$, $\varphi_3(2) = 3$, $\varphi_3(3) = 1$.

Характеристическая функция подмножества S множества U, $\chi_S\colon U \to \{0, 1\}$ определяется предписанием

$$\chi_S(x) = \begin{cases} 1, & \text{если } x \in S; \\ 0, & \text{если } x \notin S. \end{cases}$$

(П) Отображение $\mathbf{b}\colon P(U) \to \{\chi_S \mid S \in P(U)\}$, ставящее каждому подмножеству из $P(U)$ соответствующую ему характеристическую функцию, — биекция, т.е. любое подмножество множества U однозначно задается характеристической функцией и наоборот. Отображение $\mathbf{b}$ имеет областью определения $P(U)$, а т.к. $\#P(U) = 2^{\#U}$, то существует ровно $2^{\#U}$ характеристических функций.

Упорядоченная пара $\langle x, y\rangle$ - это совокупность, состоящая из двух элементов, расположенных в определенном порядке.

Две пары $\langle x, y\rangle$ и $\langle u, v\rangle$ **равны** тогда и только тогда, когда $x = u$ и $y = v$.

Упорядоченная n-ка (энка) элементов $x_1, \ldots, x_n$ обозначается $\langle x_1, \ldots, x_n\rangle$ и, по определению, есть $\langle\langle x_1, \ldots, x_{n-1}\rangle, x_n\rangle$.

Бинарным (или **двуместным**) **отношением** ρ называется множество упорядоченных пар.

Если ρ есть некоторое отношение и пара $\langle x, y\rangle$ принадлежит ему, то наряду с записью $\langle x, y\rangle \in \rho$ употребляется запись $x\rho y$. Элементы x и y называются **координатами** (или **компонентами**) отношения ρ.

n-арным (или **n-местным**) отношением называется множество упорядоченных n-ок.

Областью определения бинарного отношения ρ называется множество $D_\rho = \{x \mid \exists y\ x\rho y\}$.

Областью значения бинарного отношения ρ называется множество $R_\rho = \{y \mid \exists x\ x\rho y\}$.

(П) Пусть множество $\{\langle 1, 2\rangle, \langle 2, 4\rangle, \langle 3, 3\rangle, \langle 2, 1\rangle\}$ — бинарное отношение ρ. Тогда $D_\rho = \{1, 2, 3\}$, $R_\rho = \{1, 2, 3, 4\}$.

(П) Отношение равенства на множестве вещественных чисел $\mathbb{R}$ есть бинарное отношение ρ, состоящее из всех упорядоченных пар, в которых первый элемент равен второму и принадлежит $\mathbb{R}$. $R_\rho = D_\rho = \mathbb{R}$.

(П) Отношение "меньше" на множестве натуральных чисел $\mathbb{N}$ есть множество $\rho = \{\langle x, y\rangle \mid \exists z > 0, z \in \mathbb{N}, x + z = y\}$. $D_\rho = \mathbb{N}$, $R_\rho = \mathbb{N} \setminus \{1\}$. Для этого отношения есть специальное обозначение $<$, т.е. вместо $x\rho y$ можно писать $x < y$.

Прямым (**декартовым**) произведением множеств X и Y называется совокупность всех упорядоченных пар $\langle x, y\rangle$ таких, что $x \in X$, а $y \in Y$. Обозначение — $X \times Y$.

Каждое отношение ρ есть подмножество прямого произведения некоторых множеств X и Y таких, что $D_\rho \subset X$ и $R_\rho \subset Y$. Если $X = Y$, то говорят, что ρ — это **бинарное отношение** на X.

Прямым произведением множеств $X_1, \ldots, X_n$ называется совокупность всех упорядоченных n-ок $\langle x_1, \ldots, x_n\rangle$ таких, что $x_i \in X_i$, $i \in \{1, \ldots, n\}$. Обозначение: $X_1 \times \cdots \times X_n$. Если $X_1 = \ldots = X_n = X$, то пишут $X_1 \times \cdots \times X_n = X^n$. Любое n-местное отношение — подмножество прямого произведения некоторых множеств $X_1, \ldots, X_n$.

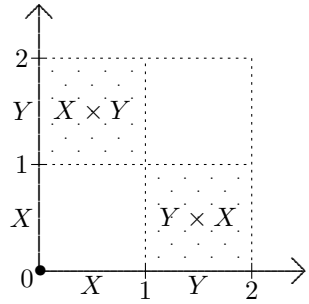
(П) Пусть $X = \{1, 2, 3\}$ и $Y = \{0, 1\}$. Тогда

$$X \times Y = \{\langle 1, 0\rangle, \langle 1, 1\rangle, \langle 2, 0\rangle, \langle 2, 1\rangle, \langle 3, 0\rangle, \langle 3, 1\rangle\},$$

$$Y \times X = \{\langle 0, 1\rangle, \langle 0, 2\rangle, \langle 0, 3\rangle, \langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle\},$$

т.е. $X \times Y \neq Y \times X$.

(П) Пусть $X = [0, 1]$, а $Y = [1, 2]$. Тогда $X \times Y$ — это множество точек квадрата $[0, 1] \times [1, 2]$ с вершинами в точках (0, 1), (0, 2), (1, 1) и (1, 2), а $Y \times X$ — это множество точек квадрата $[1, 2] \times [0, 1]$ с вершинами в точках (1, 0), (1, 1), (2, 1) и (2, 0) (см. рис.). Таким образом, $X \times Y \neq Y \times X$.



Для бинарных и n-арных отношений обычным образом определены теоретико-множественные операции объединения, пересечения и т.п. (Это следует из того, что отношения — это множества.)

(П) $= \cup <$ равно $\leqslant$.

Обратным для ρ называется отношение $\rho^{-1} = \{\langle x, y\rangle \mid \langle y, x\rangle \in \rho\}$.

Композицией отношений ρ_1 и ρ_2 называется отношение $\rho_1 \circ \rho_2 = \{\langle x, z\rangle \mid \exists y\ \langle x, y\rangle \in \rho_2$ и $\langle y, z\rangle \in \rho_1\}$.

(У10) Для любых бинарных отношений выполняются следующие свойства: 1) $(\rho^{-1})^{-1} = \rho$; 2) $(\rho_2 \circ \rho_1)^{-1} = {\rho_1}^{-1} \circ {\rho_2}^{-1}$. [1-е очевидно, 2-е в [13, стр. 12]]

(У11) Пусть f — бинарное отношение и для любых x, y и z из $\langle x, y\rangle \in f$ и $\langle x, z\rangle \in f$ следует, что $y = z$. Тогда f — **функция**, область определения которой — D_f, а $Im\, f = R_f$. [Очевидно]

Если $D_f = X$ — это область определения функции f, а $R_f \subset Y$ — это область значений f, то говорят, что f **задана** на множестве X со **значениями** во множестве Y и осуществляет отображение множества X во множество Y, $f\colon X \to Y$.

Если f — функция, то вместо $\langle x, y\rangle \in f$ обычно пишут $f\colon X \to Y$ или $y = f(x)$ и говорят, что y — значение f, **соответствующее аргументу** x, или, что y — **образ элемента** x при отображении f.

(П) Бинарное отношение $\{\langle 1, 2\rangle, \langle 2, 2\rangle, \langle ®, ©\rangle\}$ — функция, а отношение $\{\langle 1, 2\rangle, \langle 2, 4\rangle, \langle 1, 3\rangle\}$ — нет. Отношение $\{\langle x, x^2 + 2x + 1\rangle \mid x \in \mathbb{R}\}$ — функция, которую обычно обозначают $y = x^2 + 2x + 1$.

Две функции считаются **равными**, если равны определяющие их бинарные отношения.

Назовем f **n-местной функцией** из $X_1 \times \cdots \times X_n$ в Y $(n+1)$-местное отношение такое, что для любых $x_1, \ldots, x_n, y, z$ из $\langle x_1, \ldots, x_n, y\rangle \in f$ и $\langle x_1, \ldots, x_n, z\rangle \in f$ следует $y = z$. Обозначают n-местные функции как $f\colon X_1 \times \cdots \times X_n \to Y$ или $y = f(x_1, \ldots, x_n)$ и говорят, что y — значение f, соответствующее аргументам $x_1, \ldots, x_n$. Если $X_1 = \cdots = X_n = X$, то пишут $f\colon X^n \to Y$.

(У12) Композиция двух функций есть функция. При этом, если

$f: X \to Y$ и $g: Y \to Z$, то $g \circ f: X \to Z$. [Пусть $\langle x, z_1 \rangle \in g \circ f$ и $\langle x, z_2 \rangle \in g \circ f$ и пусть $f(x) = u$, тогда вследствие того, что f и g — функции, получается, что u — единственно и $g(u)$ также единственно и, следовательно, $z_1 = z_2 = g(u)$.]

(У13) Композиция двух биективных функций есть биективная функция. [Очевидно]

Пусть f^{-1} отношение обратное функции f. Если f^{-1} — функция, то она называется функцией **обратной** к f (или отображением **обратным** к f).

(У14) Отображение $f: X \to Y$ имеет обратное отображение $f^{-1}: Y \to X$ тогда и только тогда, когда f — биекция. [($\Leftarrow$) Если f — биекция, то, во-первых, f — сюръекция и, следовательно, f^{-1} определено на всем Y и, во-вторых, f — инъекция, что означает, что из $\langle y, x_1 \rangle \in f^{-1}$ и $\langle y, x_2 \rangle \in f^{-1}$ следует, что $\langle x_1, y \rangle \in f$ и $\langle x_2, y \rangle \in f$ и поэтому $x_1 = x_2$, что доказывает, что f^{-1} — функция. ($\Rightarrow$) Пусть существует f^{-1} — функция обратная к f. Тогда если f — несюръекция, то f^{-1} не определена на всем Y, т.е. f^{-1} — нефункция, что противоречит условию, следовательно, f — сюръекция, и если f — неинъекция, то существуют x_1 и x_2 такие, что $f(x_1) = f(x_2) = y$ или $f^{-1}(y) = x_1$ и $f^{-1}(y) = x_2$, что тоже противоречит тому, что f^{-1} — функция, следовательно, f — инъекция. Таким образом, доказано, что f — биекция.]

(У15) Для биективных функций f и g, заданных на множестве X верны следующие свойства:

1) $(f^{-1})^{-1} = f$;
2) $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$;
3) $f^{-1} \circ f = f \circ f^{-1} = 1_X$.

[Следуют из того, что f и g бинарные отношения.]

(У16) **Принцип Дирихле.** Пусть X — любое конечное множество. Отображение $f: X \to X$ биективно тогда и только тогда, когда f — сюръекция. [[3, стр. 35–37]]

(У17) Пусть X — любое конечное множество. Тогда множества инъективных, сюръективных и биективных отображений $f: X \to X$ совпадают. [[3, стр. 35–37]]

Специальные бинарные отношения

В математике важную роль играют два вида бинарных отношений: отношения эквивалентности и частичного порядка.

Отношение ρ на множестве X называется **рефлексивным**, если $\forall x \in X$ выполняется $x \rho x$.

Отношение ρ на множестве X называется **симметричным**, если $\forall x, y \in X \; x \rho y \; \Rightarrow \; y \rho x$.

Отношение ρ на множестве X называется **транзитивным**, если $\forall x, y, z \in X\ x\rho y$ и $y\rho z\ \Rightarrow\ x\rho z$.

Рефлексивное, симметричное и транзитивное бинарное отношение на множестве X называется **отношением эквивалентности** на этом множестве.

(П) Отношение равенства на множестве целых чисел есть отношение эквивалентности.

(П) Отношение подобия на множестве треугольников есть отношение эквивалентности.

(П) Отношение $<$ на множестве вещественных чисел нерефлексивно, несимметрично, но транзитивно.

(П) Отношение принадлежности к одной группе на множестве студентов института — это отношение эквивалентности.

Два числа $x, y \in \mathbb{Z}$ называются **сравнимыми по модулю** $n \in \mathbb{N}$, если их остатки при делении на n равны, или, что то же самое, их разность делится на n без остатка. Обозначение: $x \equiv y \pmod{n}$.

(П) Отношение сравнимости по модулю двух целых чисел есть отношение эквивалентности.

Пусть ρ — отношение эквивалентности на множестве X. **Классом эквивалентности**, порожденным элементом $x \in X$, называется подмножество множества X, состоящее из тех элементов $y \in X$, для которых $x\rho y$. Класс эквивалентности, порожденный элементом x, обозначается через $[x]$, $[x] = \{y \mid y \in X \text{ и } x\rho y\}$.

(П) Отношение равенства на множестве целых чисел порождает следующие классы эквивалентности: для любого элемента $x \in \mathbb{Z}$ $[x] = \{x\}$, т.е. каждый класс эквивалентности состоит только из одного элемента x.

(П) Отношение сравнимости по модулю n на множестве целых чисел $\mathbb{Z}$ порождает следующие классы эквивалентности: вместе с любым числом $a \in \mathbb{Z}$ в этом же классе эквивалентности содержатся все числа вида $a + kn$, где k — целое. Числа 0, 1, 2, ..., $n-1$ порождают различные классы эквивалентности, обозначаемые $[0]$, $[1]$, $[2]$, ..., $[n-1]$. Классы эквивалентности для чисел меньших нуля или больших $n-1$ совпадают с ними, так как любое целое число m можно представить в виде $m = qn + r$, где $0 \leqslant r < n, r \in \mathbb{Z}, q \in \mathbb{Z}$, т.е. $[r] = \{qn + r \mid q \in \mathbb{Z}\}$.

Классы эквивалентности, порождаемые отношением сравнимости по модулю n, называются **классами вычетов по модулю** n.

(П) Для отношения принадлежности двух студентов к одной студенческой группе классом эквивалентности является множество студентов этой группы.

(У18) Пусть ρ — отношение эквивалентности на множестве X. То-

гда: 1) если $x \in X$, то $x \in [x]$; 2) если $x, y \in X$ и $x\rho y$, то $[x] = [y]$, т. е. класс эквивалентности порождается любым своим элементом. [Для доказательства 1-й части достаточно воспользоваться рефлексивностью ρ, а для доказательства 2-й — симметричностью и транзитивностью ρ.]

(У19) Всякое разбиение множества X определяет на X отношение эквивалентности ρ: $x\rho y$ тогда и только тогда, когда x и y принадлежат одному подмножеству разбиения. [Рефлексивность, транзитивность и симметричность ρ очевидны]

(У20) Всякое отношение эквивалентности ρ на X определяет разбиение множества X на классы эквивалентности, т. е. классы эквивалентности либо не пересекаются, либо совпадают. [Следует из У18]

Совокупность классов эквивалентности элементов множества X по отношению эквивалентности ρ называется **фактор-множеством** множества X по отношению ρ и обозначается X/ρ.

(П) Для отношения принадлежности двух студентов к одной студенческой группе фактор-множеством является множество групп института.

(П) Для отношения сравнимости по модулю n фактор-множеством является множество классов вычетов по модулю n.

Бинарное отношение ρ на X называется **антисимметричным**, если $\forall x, y \in X$ из $x\rho y$ и $y\rho x$ следует $x = y$.

Рефлексивное, антисимметричное и транзитивное отношение на X называется отношением **частичного порядка** на этом множестве.

(П) Отношение $\leqslant$ на множестве вещественных чисел есть отношение частичного порядка.

(П) Отношение $\subset$ на множестве подмножеств некоторого множества есть отношение частичного порядка.

(П) Схема организации подчинения в учреждении есть отношение частичного порядка.

Множество с заданным на нем отношением частичной упорядоченности называется **частично упорядоченным**.

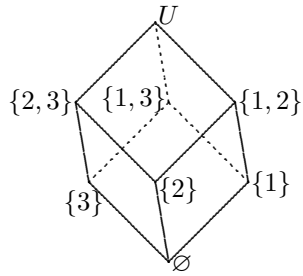
Пусть X — частично упорядоченное множество, $x, y \in X$. Элемент y **покрывает** элемент x, если $x\rho y$ и не существует такого $z \in X$, что $x\rho z$ и $z\rho y$.

Любое частично упорядоченное множество можно представить в виде схемы, в которой каждый элемент множества изображается точкой на плоскости, и если y покрывает x, то точки x и y соединяют отрезком, причем точку, соответствующую x, располагают ниже y. Такие схемы называют **диаграммами Хассе**.

(П) Пусть $U = \{1, 2, 3\}$. На множестве

$$P(U) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, U\}$$

рассмотрим отношение “быть подмножеством” (см. рис.).



(П) Пусть $U = \{1, 2, 3, 5, 6, 10, 15, 30\}$. Рассмотрим на U отношение “делит нацело”. Диаграмма Хассе для этого примера совпадет с предыдущей, если сделать замены: $\varnothing$—1, $\{1\}$—2, $\{2\}$—3, $\{3\}$—5, $\{1, 2\}$—6, $\{1, 3\}$—10, $\{2, 3\}$—15, $\{1, 2, 3\}$—30.

(П) На множестве $U = \{1, 2, 3, 5, 6, 10, 15, 30\}$ рассмотрим отношение $\leqslant$. Его диаграмма Хассе будет иметь вид вертикального отрезка прямой, на котором отмечены 8 точек, включая две крайние, помеченные разными числами от 1 (внизу) до 30 (вверху) из множества U.

Группы

Алгебраическая система с заданной на ней ассоциативной операцией называется **полугруппой**. Эту операцию далее будем называть умножением.

Полугруппа с коммутативной операцией называется **коммутативной**.

Элемент e_Π алгебраической системы Π называется **единичным**, если для $\forall \pi \in \Pi$ выполняются следующие соотношения: $\pi \times e_\Pi = e_\Pi \times \pi = \pi$.

Полугруппа с единичным элементом называется **моноидом**.

(У21) Единичный элемент в моноиде единственен. [Пусть e^*_Π — единичный и $e^*_\Pi \neq e_\Pi$, но $e^*_\Pi = e^*_\Pi e_\Pi = e_\Pi$.]

Моноид с коммутативной операцией называется **коммутативным**.

(П) Множество целых чисел — коммутативный моноид как относительно сложения, так и умножения. Обозначим эти моноиды соответственно $(\mathbb{Z}, +, 0)$, $(\mathbb{Z}, \times, 1)$.

(П) Множество целых чисел, делящихся на $n > 1$, $n \in \mathbb{N}$ — это коммутативная полугруппа (не моноид), относительно умножения. Обозначим эту полугруппу $(n\mathbb{Z}, \times)$.

(П) Пусть X — произвольное множество, а $S(X)$ — это множество всех отображений X в себя (если X — конечно, то всего таких отображений ровно $\#X^{\#X}$). Тогда относительно операции композиции отображений, $\circ$, $S(X)$ — некоммутативный моноид $(S(X), \circ, 1_X)$.

(П) Пусть $A = \{a_1, \ldots, a_n\}$ — конечное множество символов — алфавит. Конечная последовательность таких символов — это слово в алфавите A. На множестве Π всех слов в алфавите A введем бинарную операцию "приписывание", т. е. если $\pi_1 = a_{11} \cdots a_{1k} \in \Pi$, $\pi_2 = a_{21} \cdots a_{2l} \in \Pi$, то $\pi_1\pi_2 = a_{11} \cdots a_{1k}a_{21} \cdots a_{2l}$. Тогда Π — это некоммутативная полугруппа (Π,"приписывание"), называемая свободной полугруппой.

Подмножество Π^* полугруппы Π называется **подполугруппой**, если $\forall \pi_1, \pi_2 \in \Pi^*$, $\pi_1\pi_2$ и $\pi_2\pi_1$ также принадлежит Π^*. В этом случае также говорят, что подмножество Π^* замкнуто относительно рассматриваемой операции.

(У22) Любая подполугруппа является полугруппой. [Очевидно]

Если M — моноид и $M^* \subset M$ — подполугруппа, содержащая единичный элемент, то M^* называется **подмоноидом**.

(У23) Любой подмоноид является моноидом. [Очевидно]

(П) Множество целых чисел, кратных $n \in \mathbb{N}$, — подполугруппа в полугруппе ($\mathbb{Z}$,$\times$,1) и подмоноид в ($\mathbb{Z}$,+,0).

(П) В полугруппе Π всех слов в алфавите A подмножество всех слов в алфавите $A^* \subset A$ — подполугруппа Π.

Элемент a моноида M называется **обратимым**, если существует $b \in M$ такой, что $ab = ba = e_M$. Элемент b называется **обратным** a и обозначается a^{-1}.

(У24) Обратный элемент единственен. [Пусть существует обратный элемент $b \neq a^{-1}$, но тогда $b = be_M = baa^{-1} = a^{-1}$.]

(У25) $(a^{-1})^{-1} = a$. [Элемент $(a^{-1})^{-1}$ является обратным для a^{-1}, но обратным к a^{-1} является единственный элемент a.]

(У26) $(ab)^{-1} = b^{-1}a^{-1}$ [Элемент $(ab)^{-1}$ является обратным для ab по определению. Докажем, что для ab обратным также является элемент $b^{-1}a^{-1}$. Это следует из следующих соотношений $b^{-1}a^{-1}ab = b^{-1}e_Mb = b^{-1}b = e_M$ и $abb^{-1}a^{-1} = e_M$. Для завершения доказательства используется утверждение об единственности обратного элемента.]

(У27) Множество всех обратимых элементов моноида образует подмоноид. [Пусть M — моноид, а $M^* \subset M$ — множество всех обратимых элементов в M, тогда нужно доказать, что $\forall a, b \in M^*$, $(ab)^{-1} \in M$. Последнее следует из того, что $(ab)^{-1} = b^{-1}a^{-1}$.]

Пусть $S = \{s_1, \ldots, s_n\}$ — подмножество элементов полугруппы Π такое, что любой элемент из Π может быть представлен как произве-

дение элементов из S, т. е. $\forall \pi \in \Pi\ \pi = s_{\pi_1} \cdots s_{\pi_m}$, $\forall \pi_i$ $(1 \leqslant \pi_i \leqslant n$, $1 \leqslant i \leqslant m)$. Тогда множество S называется **множеством образующих** полугруппы Π.

(П) Для полугруппы (П, "приписывание") всех слов из алфавита $A = \{a_1, \ldots, a_n\}$, само множество A является множеством образующих.

(П) Для полугруппы $(\mathbb{Z},+,0)$ множеством образующих является множество $\{-1, 1\}$.

(П) Для полугруппы $(\mathbb{Z},\times,1)$ множеством образующих является множество всех простых чисел, -1 и 0.

Алгебраическая система, состоящая из непустого множества G и одной бинарной операции, называется **группой**, если выполняются следующие условия:

1) операция в G ассоциативна;
2) в G существует единичный элемент e_G: $\forall g \in G \quad e_G g = g e_G = g$;
3) $\forall g \in G \quad \exists g^{-1} \in G \quad g g^{-1} = g^{-1} g = e_G$.

Иными словами, **группа** — это моноид, все элементы которого обратимы.

Если операция в группе коммутативна, то такую группу называют **коммутативной** или **абелевой**.

Подмножество $H \subset G$ называется **подгруппой** G, если H является группой относительно определенной для G операции.

(П) $(\mathbb{N}, +)$ — это коммутативная полугруппа; $(\mathbb{Z}, +, 0)$, $(\mathbb{Q}, +, 0)$, $(\mathbb{R}, +, 0)$, $(\mathbb{C}, +, 0)$ — коммутативные группы.

(П) $(\mathbb{N}, \times, 1)$ и $(\mathbb{Z}, \times, 1)$ — коммутативные моноиды, а $(\mathbb{Q}\backslash\{0\}, \times, 1)$, $(\mathbb{R}\backslash\{0\}, \times, 1)$, $(\mathbb{C}\backslash\{0\}, \times, 1)$ — коммутативные группы.

(П) $(\mathbb{Q}_+, +, 0)$ подгруппа группы $(\mathbb{Q}, +, 0)$.

(П) Пусть X — произвольное множество, а $B(X)$ — это множество всех биекций X в себя. Тогда $B(X)$ — группа относительно операции композиции отображений, $\circ$. Это, так называемая, группа преобразований $(B(X), \circ, 1_X)$.

(П) Рассмотрим множество M_n квадратных матриц размерности $n \times n$ с определителем, отличным от нуля. Это некоммутативная группа $(M_n, \times, E_n)$.

(П) Рассмотрим множество классов вычетов по модулю n: [0], [1], …, $[n-1]$, где $a \in [k]$ тогда и только тогда, когда $a \equiv k \pmod{n}$, где $0 \leqslant k < n$. Множество классов вычетов образует группу $G = (\mathbb{Z}_n, +, [0])$ относительно операции сложения классов, определяемой по закону $[k] + [l] = [r]$, где $(k + l) \equiv r \pmod{n}$ и $0 \leqslant r < n$, $e_G = [0]$ и $[k]^{-1} = [n - k]$. При $n = 3$ эту группу можно задать следующей таблицей:

$+$	$[0]$	$[1]$	$[2]$
$[0]$	$[0]$	$[1]$	$[2]$
$[1]$	$[1]$	$[2]$	$[0]$
$[2]$	$[2]$	$[0]$	$[1]$

.

Таблица, определяющая операцию группы, называется **таблицей Кэли**.

Если группа состоит из конечного числа элементов, то она называется **конечной**, а число ее элементов — **порядком** группы. В противном случае группа называется **бесконечной**.

Циклические группы. Группы подстановок

(У28) Пусть G — группа, а F и H — ее подгруппы. Тогда пересечение F и H, $D = F \cap H$, также является подгруппой группы G. [D — непусто, т. к. оно содержит единичный элемент. Если $a, b \in D$, то $a, b \in F$ и $a, b \in H$ и, следовательно, $a^{-1}, b^{-1}, ab \in F$ и $a^{-1}, b^{-1}, ab \in H$, откуда $a^{-1}, b^{-1}, ab \in D$.]

Следующее утверждение доказывается методом **математической индукции**, который состоит из двух этапов:

1) доказывается (базис индукции), что утверждение верно для некоторого целого k;
2) доказывается для любого $n \geqslant k$, что из предположения о верности утверждения для n следует, что утверждение верно и при $n + 1$.

Если удастся пройти оба этапа, то это доказывает верность утверждения для любого $n \geqslant k$.

(У29) Пересечение любого числа подгрупп некоторой группы есть подгруппа. [Пусть число подгрупп — k. В предыдущем утверждении доказано, что при $k = 2$ утверждение верно. Пусть утверждение верно для $k = n$. Рассмотрим случай $k = n+1$. Пересечение n из $n+1$ подгрупп будет подгруппой по предположению, но пересечение этой подгруппы с одной оставшейся будет подгруппой согласно доказанному базису индукции.]

Пусть S — произвольное непустое подмножество группы G. Рассмотрим все возможные подгруппы G, которые содержат S в качестве подмножества. Одной из них будет, в частности, сама группа G. В силу предыдущего утверждения пересечение всех таких подгрупп будет подгруппой группы G, которая называется подгруппой, **порожденной** множеством S, и обозначается $\langle S \rangle$.

Если множество S состоит из одного элемента a, то порожденная им подгруппа называется **циклической**, порожденной элементом a, и обозначается $\langle a \rangle$.

Введем следующие обозначения $\forall a \in G\ (a^{-1})^k = a^{-k}$ и $a^0 = e_G$.

(У30) Циклическая подгруппа $\langle a \rangle$, состоит из всех степеней элемента a. [[13, стр. 107]]

Группа, совпадающая с одной из своих циклических подгрупп, называется **циклической**, а элемент, из степеней которого состоит циклическая группа, — ее **образующим**.

(У31) Всякая циклическая группа коммутативна. [Любой элемент такой группы — это степень ее образующего элемента a и, следовательно, $a^n a^m = a^{n+m} = a^m a^n$.]

(П) Группа $(\mathbb{Z}, +, 0)$ — циклическая, ее образующий элемент 1 (или -1).

(У32) Всякая подгруппа циклической группы является циклической. [[13, стр. 108]]

Пусть G — группа, a — некоторый ее элемент. Если все степени элемента a различны, то говорят, что a имеет **бесконечный** порядок. Если же для некоторых $n < m$ (m, $n \in \mathbb{Z}$), $a^m = a^n$, то, домножив обе части равенства на a^{-n} получим, что $a^{m-n} = e_G$, т.е. существует $k = m - n > 0$ такой, что $a^k = e_G$. Пусть $q \in \mathbb{N}$ — наименьшее число, для которого $a^q = e_G$. Тогда говорят, что a — элемент **конечного** порядка q.

(П) Пусть G группа классов вычетов по модулю 3, $(\mathbb{Z}_3, +, [0])$. Тогда $[1]^4 = [1]^7$.

(П) Порядок элемента 1 (-1) в группе $(\mathbb{Z}, +, 0)$ бесконечен, а порядок элемента $[1]$ в группе $(\mathbb{Z}_3, +, [0])$ равен 3.

Пусть X — конечное множество из n элементов. Группа всех биекций множества X в себя, $(B(X), \circ, 1_X)$, называется **симметрической** группой степени n. Без ограничения общности можно считать, что $X = \{1, 2, \ldots, n\}$. Каждая биекция $\varphi\colon X \to X$ из $B(X)$ называется **подстановкой** и записывается как $\begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}$, где под элементом k ($1 \leqslant k \leqslant n$) записан его образ $\varphi(k) = i_k$. Единичную (тождественную) подстановку 1_X обозначим e. Симметрическая группа степени n обозначается S_n.

(У33) Симметрическая группа степени n содержит $n!$ элементов, $\#B(X) = n!$. [Очевидно]

Произведением подстановок φ и ψ называется их композиция $(\varphi\psi)(k) = \varphi(\psi(k))$.

(П) Для подстановок $\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}$ и $\psi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$, $\psi\varphi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$ и $\varphi\psi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$, т.е. $\psi\varphi \neq \varphi\psi$.

(П) Группа S_3 состоит из шести элементов: $e = a_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$, $a_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$, $a_3 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$, $a_4 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$, $a_5 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$, $a_6 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$.

(П) Рассмотрим подстановку $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 4 & 3 & 5 & 2 \end{pmatrix}$, следовательно,

$$\pi^2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 3 & 2 & 4 \end{pmatrix}, \quad \pi^3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} = e,$$

т.е., например, $\pi(2) = 4$, $\pi^2(2) = 5$, $\pi^3(2) = 2$.

В последнем примере в подстановке π элементы 1 и 3 остаются на месте, элемент 2 переходит в 4, элемент 4 — в 5, а элемент 5 — снова в 2. Такая подстановка называется **циклом** (245) **длины** 3. Этот же цикл можно записать и как (452) или (524).

В общем случае подстановка π, перемещающая $j_1, j_2, \ldots, j_k$ так, что $\pi(j_1) = j_2, \ldots, \pi(j_{k-1}) = j_k, \pi(j_k) = j_1$ называется **циклом длины** k и обозначается $(j_1\, j_2 \cdots j_k)$.

Циклы называются **независимыми**, если они не имеют общих переставляемых элементов.

(П) Циклы (123) и (45) — независимы, а циклы (234) и (125) — зависимы.

Циклы длины 1 в записи подстановки можно опускать, например,

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 2 & 8 & 7 & 6 & 1 & 4 & 3 \end{pmatrix} = (156)(2)(38)(47) = (156)(38)(47).$$

Два элемента i и j множества X называются **эквивалентными** относительно подгруппы H группы $G = B(X)$, если $j = \pi(i)$ для некоторого элемента $\pi \in H$.

(У34) Введенное отношение есть отношение эквивалентности на множестве X. [Его рефлексивность, симметричность и транзитивность очевидны].

Это отношение разбивает множество X на классы эквивалентности по этому отношению, называемые **орбитами**. Обозначение — Hx — орбита для $x \in X$ по H. Множество всех орбит естественно обозначить X/H.

(П) Рассмотрим $H = \{e, (12)\} \subset S_3$, $X = \{1, 2, 3\} = \{1, 2\} \cup \{3\}$, т.е. получаем две орбиты $H1 = H2$ и $H3$.

(П) При $H = \{e, (135)(246), (153)(264)\} \subset S_6$, $X = \{1, 2, 3, 4, 5, 6\}$ $H1 = H3 = H5 = \{1, 3, 5\}$ и $H2 = H4 = H6 = \{2, 4, 6\}$ — две орбиты.

(У35) Каждая подстановка в S_n может быть представлена произведением независимых циклов. Разложение подстановки в произведение независимых циклов определено однозначно с точностью до порядка множителей. [[13, стр. 109]]

Цикл длины 2 называется **транспозицией**.

(У36) Любой цикл можно представить в виде произведения транспозиций. [Это следует из следующего равенства $(i, \pi(i), \ldots, \pi^{s-1}(i)) = (i, \pi^{s-1}(i))(i, \pi^{s-2}(i)) \cdots (i, \pi(i))$ при $\pi^s = e$]

(П) В группе S_4 $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix} = (123) = (13)(12) = (13)(24)(12)(14)$.

Декрементом (D) подстановки называется разность между числом её элементов и числом независимых циклов её разложения. Декремент — это наименьшее число транспозиций, в которое раскладывается подстановка.

Числом инверсий (I) подстановки назовём сумму количеств меньших элементов, стоящих правее. **Инверсия** — это пара элементов подстановки, нарушающая порядок от меньшего к большему. Число инверсий в обратной подстановке равно числу инверсий в исходной.

Чётность (T) подстановки — это чётность числа транспозиций, в произведение которых можно разложить подстановку. Поизведение двух нечётных или двух чётных подстановок чётно, а произведение четной подстановки на нечётную — нечётно. Число инверсий как и декремент определяет чётность.

(П) $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix} = (123)(4) = (231)$, $D(\pi) = 4 - 2 = 2$, $I(\pi) = 0 + 1 + 1 + 0 = 2$, $T(\pi)$ — чётно.

(П) Определитель матрицы M размерности $n \times n$ с элементами a_{ij} можно вычислить по формуле $\det M = \sum_{\pi \in S_n} (-1)^{T(\pi)} f(\pi)$, где $\pi = \begin{pmatrix} 1 & \ldots & n \\ \pi(1) & \ldots & \pi(n) \end{pmatrix}$, а $f(\pi) = a_{1\pi(1)} \times \cdots \times a_{n\pi(n)}$.

Для каждого $x \in X$ множество $G_x = \{g \in G \mid gx = x\}$, образует подгруппу G, называемую **стабилизатором**.

(П) Рассмотрим $H = \{e, (12)\} = \langle (12) \rangle \subset S_3$, $H_1 = H_2 = \{e\}$, $H_3 = H$.

(П) Рассмотрим саму $G = S_3$, $G_1 = \langle (23) \rangle$, $G_2 = \langle (13) \rangle$, $G_3 = \langle (12) \rangle$.

(П) При $H = \{e, (135)(246), (153)(264)\} \subset S_6$ $H_1 = \ldots = H_6 = \{e\}$.

Группы G и H называются **изоморфными**, если существует биекция $\varphi \colon G \to H$, сохраняющая групповую операцию, т. е. $\forall g_1, g_2 \in G$ $\varphi(g_1 g_2) = \varphi(g_1)\varphi(g_2)$.

(П) Группа положительных вещественных чисел относительно опе-

рации умножения изоморфна группе вещественных чисел относительно операции сложения: биекция $\varphi(x) = \ln x$ устанавливает требуемый изоморфизм, т.к. $\ln xy = \ln x + \ln y$.

(У37) Изоморфизм переводит единичный элемент в единичный, т.е. для изоморфных групп H и G с биекцией $\varphi : G \to H$ $\varphi(e_G) = e_H$. [Возьмем произвольный $g \in G$, тогда $\varphi(g) = \varphi(e_G g) = \varphi(e_G)\varphi(g) = \varphi(g e_G) = \varphi(g)\varphi(e_G)$, т.е. $\varphi(e_G) = e_H$.]

(У38) Изоморфизм переводит обратный элемент в обратный, т.е. для изоморфных групп H и G с биекцией $\varphi: G \to H$ $\forall g \in G$ $\varphi(g^{-1}) = \varphi(g)^{-1}$. [$\varphi(e_G) = \varphi(g^{-1}g) = \varphi(g^{-1})\varphi(g) = \varphi(gg^{-1}) = \varphi(g)\varphi(g^{-1}) = e_H$, т.е. $\varphi(g^{-1}) = \varphi(g)^{-1}$.]

(У39) Отображение обратное изоморфизму существует и является изоморфизмом. [Пусть $\varphi: G \to H$ — изоморфизм, тогда существование $\varphi^{-1}: H \to G$ следует из того, что φ — биекция. Пусть h_1, $h_2 \in H$. Докажем, что $\varphi^{-1}(h_1 h_2) = \varphi^{-1}(h_1)\varphi^{-1}(h_2) = g_1 g_2$, где элементы $g_1 = \varphi^{-1}(h_1) \in G$ и $g_2 = \varphi^{-1}(h_2) \in G$: $\varphi^{-1}(h_1 h_2) = \varphi^{-1}(\varphi(g_1)\varphi(g_2)) = \varphi^{-1}(\varphi(g_1 g_1)) = g_1 g_2$.]

(У40) Все циклические группы одного порядка изоморфны. [[13, стр. 111–112]]

(П) Группа классов вычетов по модулю n, $(\mathbb{Z}_n, +, [0])$, изоморфна любой циклической группе порядка n.

(П) Группа классов вычетов по модулю 6, $\mathbb{Z}_6 = \{[0], [1], [2], [3], [4], [5]\}$, и симметрическая группа $S_3 = \{e, (12), (13), (23), (123), (132)\}$ неизоморфны, т.к. в 1-й группе есть только один элемент $x = [3]$, отличный от единичного, такой, что $xx = e$; во 2-й группе таких элементов три: (12), (13), (23). Таким образом, группы одного порядка могут быть неизоморфны.

(У41) **Теорема Кэли.** Всякая конечная группа порядка n изоморфна некоторой подгруппе симметрической группы S_n. [[13, стр. 112–113]]

(П) Группа классов вычетов по модулю 6, $(\mathbb{Z}_6, +, [0])$, изоморфна подгруппе симметрической группы $H \subset S_6$,

$$H = \{e, (123456), (135)(246), (14)(25)(36), (153)(264), (165432)\} = \langle (123456) \rangle.$$

Пусть H — подгруппа группы G. **Левым смежным классом** G по H называется множество gH всех элементов вида gh, где g — фиксированный элемент из G (он определяет смежный класс), а h — любой элемент из H, т.е. $gH = \{gh \mid h \in H\}$. **Правый смежный класс** определяется аналогично: $Hg = \{hg \mid h \in H\}$.

(П) Если взять $g = e_G$, то получится, что сама подгруппа H является смежным (как левым, так и правым) классом G.

(У42) Левые (правые) смежные классы G по H либо не пересекаются, либо совпадают, т.е. множество левых (правых) смежных классов

образует разбиение G, определяющее соответствующее отношение эквивалентности. [Предположим, что классы g_1H и g_2H имеют общий элемент $a = g_1h_1 = g_2h_2$. Тогда $g_2 = g_1h_1{h_2}^{-1}$ и любой элемент g_2h имеет вид $g_1h_1{h_2}^{-1}h$, где $h_1{h_2}^{-1}h \in H$. Значит, $g_2H \subset g_1H$. Аналогично получается, что $g_1H \subset g_2H$, и, следовательно, $g_1H = g_2H$.]

(У43) Два элемента $g_1, g_2 \in G$ лежат в одном левом (правом) смежном классе G по H тогда и только тогда, когда $g_2^{-1}g_1 \in H$. [($\Rightarrow$) Пусть $g_1 \in gH$ и $g_2 \in gH$, т. е. $\exists h_1, h_2 \in H$ такие, что $g_1 = gh_1$ и $g_2 = gh_2$. Тогда $g_2^{-1}g_1 = (gh_2)^{-1}gh_1 = h_2^{-1}g^{-1}gh_1 = h_2^{-1}h_1 \in H$. ($\Leftarrow$) Пусть $g_2^{-1}g_1 \in H$. Тогда $g_1 = (g_2g_2^{-1})g_1 = g_2(g_2^{-1}g_1) \in g_2H$, $g_2 = g_2e_G \in g_2H$. Для правого смежного класса доказательство аналогично.]

Множество левых (правых) смежных классов G по H обозначается G/H. Это множество классов эквивалентности по отношению принадлежности к одному левому (правому) смежному классу или фактормножество. Если обозначить отношение принадлежности к одному левому (правому) смежному классу через ρ, то $G/H = G/\rho$.

Разбиение G на классы эквивалентности по подгруппе $H \subset G$ образует орбиты.

(У44) **Теорема Лагранжа.** Порядок любой конечной группы G делится на порядок любой ее подгруппы H так, что $\#G = \#(G/H)\#H$. [Из У42 вытекает, что G — объединение непересекающихся левых смежных классов G по H. Каждый из смежных классов содержит $\#H$ элементов. Поэтому $\#G = \#(G/H)\#H$.]

(П) Множества левых и правых смежных классов группы по одной и той же подгруппе, вообще говоря, различны. Пусть $S_3 = \{e, (12), (13), (23), (123), (132)\}$ — симметрическая группа степени 3, а H — подгруппа, порожденная элементом (12), $H = \langle(12)\rangle = \{e, (12)\}$. Тогда S_3 разбивается на следующие левые смежные классы по H: $\{e, (12)\}$, $\{(13), (123)\}$, $\{(23), (132)\}$. Правые смежные классы S_3 по H следующие: $\{e, (12)\}$, $\{(13), (132)\}$, $\{(23), (123)\}$.

Подгруппа H группы G называется **нормальной**, если множества левых и правых смежных классов G по H совпадают, т. е. $\{gH \mid g \in G\} = \{Hg \mid g \in G\}$. Последнее означает, что для всякого элемента g из G и для всякого элемента h из H можно подобрать такие элементы h_1 и h_2 из H, что $gh = h_1g$ и $hg = gh_2$.

(У45) Любая подгруппа коммутативной группы — нормальна. [Следует из предыдущего определения, если в качестве $h_1 = h_2 = h$.]

(У46) **Теорема об орбитах и стабилизаторах.** $\#Hx = \#(H/H_x) = \#(H)/\#(H_x)$. [Последнее равенство — это теорема Лагранжа. [1, стр. 19]]

Множеством **фиксируемых** элементом $g \in G$ точек назовём $X_g =$

$\{x \in X \mid gx = x\}$.

(П) Рассмотрим $G = S_3$, $X_e = X$, $X_{(12)} = \{3\}$, $X_{(13)} = \{2\}$, $X_{(23)} = \{1\}$, $X_{(123)} = X_{(132)} = \varnothing$.

(П) При $G = S_6$ $X_e = X$, $X_{(12)} = \{3, 4, 5, 6\}$, $X_{(123)} = \{4, 5, 6\}$, $X_{(1235)} = \{4, 6\}$, $X_{(12345)} = \{6\}$, $X_{(135)(246)} = \varnothing$, ...

(У47) **Лемма Бёрнсайда (Кочи-Фробениуса).** Число орбит по группе H равно среднему арифметическому количеств фиксируемых элементами группы точек, т.е. $\#(X/H) = \frac{1}{\#H} \sum\limits_{h \in H} \#X_h$. [[1, стр. 20]]

(П) Рассмотрим $H = \{e, (12)\} \subset S_3$, $\#X_e = 3$, $\#X_{(12)} = 1$, $\#H = 2$, $\frac{1}{\#H} \sum_{h \in H} \#X_h = \frac{1}{2}(3 + 1) = 2$.

(П) При $H = \{e, (135)(246), (153)(264)\} \subset S_6$ $\#H = 3$, $\#X_e = 6$, $\#X_{(135)(246)} = \#X_{(153)(264)} = 0$, $\frac{1}{\#H} \sum_{h \in H} \#X_h = \frac{1}{3}(6 + 0 + 0) = 2$.

Произведением двух подмножеств X и Y группы G называются все элементы G вида xy, где $x \in X$ и $y \in Y$, т.е. $XY = \{xy \mid x \in X, y \in Y\}$.

(У48) Произведение смежных классов по нормальной подгруппе является смежным классом по этой же подгруппе. $[(g_1H)(g_2H) = \{g_1h_1g_2h_2 \mid h_1, h_2 \in H\} = [\forall g_2, h_1\ \exists {h_1}'\ h_1g_2 = g_2{h_1}'] = \{g_1g_2{h_1}'h_2 \mid {h_1}', h_2 \in H\} = \{g_1g_2h_3 \mid h_3 \in H\} = g_1g_2H$, т.е. произведение левых смежных классов, соответствующих элементам g_1 и g_2, есть левый смежный класс, соответствующий элементу g_1g_2. Доказательство для правых смежных классов аналогично.]

(У49) Если H — нормальная подгруппа группы G, то G/H — группа относительно операции произведения подмножеств. [Если взять в качестве единичного элемента подгруппу H, а в качестве элемента обратного элементу gH — смежный класс $g^{-1}H$, то множество G/H с операцией произведение подмножеств удовлетворяет определению группы.]

Пусть G — группа, H — ее нормальная подгруппа. Тогда группа G/H называется **фактор-группой** группы G.

(П) Пусть $G = (\mathbb{Z}, +, 0)$ — аддитивная группа целых чисел, а $n\mathbb{Z}$ — подгруппа G чисел кратных n. $n\mathbb{Z}$ — нормальна вследствие коммутативности G. Тогда $G/n\mathbb{Z}$ — циклическая группа порядка n. Она изоморфна группе классов вычетов по модулю n, $\mathbb{Z}_n$.

(У50) Фактор-группа абелевой группы — абелева. $[(g_1H)(g_2H) = g_1g_2H = g_2g_1H = (g_2H)(g_1H)$.]

(У51) Фактор-группа циклической группы — циклическая. [Пусть $G = \langle a \rangle$, тогда $\forall g \in G\ \exists m \in \mathbb{Z}\ a^m = g$ и $gH = a^mH = (aH)^m$, т.е. $G/H = \langle aH \rangle$.]

Для любого элемента $g \in G$ и некоторого $f \in G$, называемого **трансформационным**, $\overline{g} = f^{-1}gf$ называется **сопряженным** g, а

операция получения $\overline{g}$ — **сопряжением**.

(У52) Пары всех сопряжённых элементов образуют отношение эквивалентности. [Очевидно]

Классы эквивалентности (орбиты) по этому отношению называют **сопряжёнными** классами G.

Подгруппы группы G образуют **узлы**, а отношение частичного порядка $\subset$ — связи между узлами. Построенная по этому отношению диаграмма Хассе называется **решёткой** и обозначается $S(G)$. Подгруппы G и e называют **полюсами** решётки.

(П) S_3; $(\mathbb{Z}_p, +, [0])$, где p — простое; $\mathbb{Z}_6$.

Группы небольших порядков

Тождественный элемент образует единственную группу первого порядка, C_1. Её примерами будут $(0, +, 0)$ или $(1, \cdot, 1)$. Циклические группы порядка n будем обозначать C_n. Примерами циклических групп для любого порядка n будут $(\mathbb{Z}_n, +, [0]) = \langle [1] \rangle$.

Примерами единственной группы второго порядка C_2 будут $(\{-1, 1\}, \cdot, 1)$, $(\mathbb{Z}_3 \setminus [0], \cdot, [1])$, $(\{0, 1\}, \mathrm{XOR}, 0)$.

Существует только одна группа порядка 3, C_3. Её примерами могут быть группа перестановок $\{e,\ (123),\ (132)\}$, группа матриц $\{E, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\}$ относительно умножения.

Группа, порядок которой — простое число, — это всегда единственная циклическая группа. Поэтому существует только одна группа порядка 5, 7, 11, ...

Одна из групп порядка 4, C_4, — это, например, подстановки $\{e,\ (1234),\ (13)(24),\ (1432)\} = \langle (1234) \rangle$, единицы комплексных чисел относительно умножения $(\{1, -1, i, -i\}, \cdot, 1) = \langle i \rangle$, группа матриц $\left\langle \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\rangle$ относительно умножения.

Другая группа порядка 4, $C_2 \times C_2 = C_2^2$ — это, например, подстановки $\{e, (12), (34), (12)(34)\} = \{e, (12)\} \cdot \{e, (34)\} = \langle (12), (34) \rangle$.

Одной из групп порядка $n!$ всегда будет симметрическая группа степени n, S_n.

Поэтому одна из групп порядка 6 — это S_3 — некоммутативная группа наименьшего порядка. Другая группа порядка 6 — это $C_2 C_3$, примером которой могут быть подстановки $\{e,\ (12),\ (345),\ (12)(345),\ (354), (12)(354)\} = \{e, (12)\} \cdot \{e, (345), (354)\}, = \langle (12), (345) \rangle$.

Всего существует 5 групп порядка 8. Одна из них, Q, — это группа базисных единиц кватернионов, т.е. чисел вида $a + bi + cj + dk$, где a, b, c, $d \in \mathbb{R}$. Действия с мнимыми единицами i, j, k определяется по

правилам

$\times$	i	j	k
i	-1	k	$-j$
j	$-k$	-1	i
k	j	$-i$	-1

Группа базисных единиц кватернионов, ($\{1, -1, i, -i, j, -j, k, -k\}$, $\cdot$, 1), изоморфна группе подстановок $\langle(1234)(5678), (1638)(2547)\rangle$, если сопоставить 1 e, i (1234)(5678), j (1638)(2547) и $k = ij$ (1735)(2648). Очевидно, что $i^4 = 1$, $i^2 = j^2$, $iji = j$.

Другая группа порядка 8, C_2C_4, — это например, $\langle(12), (3456)\rangle$, т.е. она порождается двумя подстановками, циклами длины 2 и 4. Если обозначить эти подстановки a и b, то, очевидно, что $a^2 = b^4 = e$, $ab = ba$.

Следующая группа 8-го порядка, C_2^3, порождается тремя подстановками-циклами длины 2, например, $\langle(12), (34), (56)\rangle$.

Последняя группа порядка 8 — это группа диэдра, D_4, или группа симметрий правильного n-угольника. Существуют симметрии вращения и отражения относительно биссектрис углов. Для квадрата с вершинами 1, 2, 3, 4 вращения задаются группой $\langle(1234)\rangle$, а отражения — группой $\langle(13), (24)\rangle$. Но (1234)(1234)(13) = (24), поэтому, примером D_4 может быть $\langle(1234), (13)\rangle$. Если обозначить порождающие элемента за a и b, то, очевидно, что $a^4 = b^2 = e$, $aba = b$.

Из двух групп порядка 9, одна — это C_3^2, например, $\langle(123), (456)\rangle$. Если обозначить выбранные подстановки a и b, то, очевидно, что $a^3 = b^3 = e$, $ab = ba$.

Циклическая группа порядка 10, C_{10}, может быть представлена в виде C_2C_5. Её примером может быть $\langle(12), (34567)\rangle = \langle(12)(34567)\rangle$.

Кроме неё ещё существует только одна группа 10-го порядка, D_5 — группа симметрий правильного 5-угольника. Её примером может быть $\langle(12345), (12)(35)\rangle$. Если обозначить выбранные подстановки a и b, то $a^5 = b^2 = e$, $aba = b$.

Чем больше способов разложить число на множители, тем, как правило, большим будет количество групп этого порядка, например, количество групп порядка 16 — 14.

Кольца и поля

Алгебраическая система, состоящая из непустого множества K и двух определенных на нем бинарных операций, называемых сложением ($+$) и умножением ($\times$), называется **кольцом** $(K, +, \times)$, если выполнены следующие условия: 1) относительно сложения K — коммутативная группа; 2) относительно умножения K — полугруппа; 3) $\forall a, b, c \in K$ верны законы дистрибутивности $a(b + c) = ab + ac$ и $(a + b)c = ac + bc$.

Если операция умножения коммутативна в K, то K называют **коммутативным** кольцом.

Если относительно умножения K — моноид, то K называют **кольцом с единицей**.

Порядком кольца называют количество элементов в нем.

Подмножество L кольца K называется **подкольцом**, если L — коммутативная группа относительно сложения и полугруппа относительно умножения.

(У53) Подкольцо является кольцом. Пересечение подколец — это кольцо. [Очевидно]

Подкольцом, **порожденным** множеством $S \subset K$, называется пересечение всех подколец K, содержащих S.

(П) Множество целых чисел относительно операций сложения и умножения — коммутативное кольцо с единицей, $(\mathbb{Z}, +, \times)$. Множество $n\mathbb{Z}$ целых чисел, делящихся на $n > 1$, — коммутативное подкольцо кольца $(\mathbb{Z}, +, \times)$ без единицы.

(П) Множество $\mathbb{Z}_n$ классов вычетов по модулю n с операциями сложения (совпадает с операцией $+$ для группы $(\mathbb{Z}_n, +, [0])$) и умножения (определяется правилами $[k][l] = [r]$, где $kl \equiv r \pmod{n}$ и $0 \leqslant r < n$) является коммутативным кольцом $(\mathbb{Z}_n, +, \times)$ с единицей.

Пусть K — произвольное коммутативное кольцо. Рассмотрим всевозможные многочлены $a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$, где $n \in \mathbb{Z}_+$, а $a_0,\ a_1,\ a_2,\ \ldots,\ a_n \in K$. Множество таких многочленов относительно операций сложения и умножения многочленов образует коммутативное кольцо. Оно называется **кольцом многочленов** от переменной x над кольцом K и обозначается $K[x]$.

(П) В качестве K для кольца многочленов можно взять кольцо целых, рациональных или вещественных чисел или кольцо классов вычетов по модулю n.

Обозначим через 0_K единичный элемент кольца K относительно сложения — этот элемент называется **нулевым**, и обозначим через $-b$ элемент кольца K обратный b относительно сложения.

(У54) Имеют место следующие свойства кольца K, $\forall a, b \in K$:

1) $0_K a = a0_K = 0_K$ [$aa = a(a + 0_K) = aa + a0_K =$
$= a(0_K + a) = a0_K + aa \ \Rightarrow\ a0_K = 0_K$ и
$aa = (a + 0_K)a = aa + 0_K a =$
$= (0_K + a)a = 0_K a + aa \ \Rightarrow\ 0_K a = 0_K$];
2) $(-a)b = a(-b) = -ab$ [$0_K = a0_K = a(b + (-b)) = ab + a(-b) \Rightarrow$ $a(-b) = -ab$ и т.д.];
3) (только для колец с единицей) $-a = (-e_K)a$. [Следует из второго свойства: $-a = e_K(-a) = (-e_K)a$].

Если $a \neq 0_K$ и $b \neq 0_K$ элементы кольца K такие, что $ab = 0_K$, то a и b называются **делителями нуля**.

Если в кольце нет делителей нуля, то такое кольцо называется **кольцом без делителей нуля**.

(П) Кольца целых, рациональных и вещественных чисел — это кольца без делителей нуля.

(П) Кольцо классов вычетов по модулю n — это кольцо с делителями нуля в случаях, когда $\exists k, l \in \mathbb{Z}\ kl = n$, $1 < k < n$, $1 < l < n$, т. е. когда n — составное; в этом случае $[k]$ и $[l]$ — делители нуля. Например, в кольце классов вычетов по модулю 4 $[2]$ — делитель нуля.

Пусть K — кольцо с единицей. Элемент a из K называется **обратимым**, если существует элемент a^{-1} из K такой, что $aa^{-1} = a^{-1}a = e_K$.

(У55) Обратимый элемент не может быть делителем нуля. [Пусть a — обратимый элемент K, тогда если $ab = 0_K$, то $b = e_K b = a^{-1}ab = a^{-1}0_K = 0_K$.]

(У56) Все обратимые элементы кольца образуют группу относительно умножения. [Все обратимые элементы кольца относительно умножения образуют моноид, который является группой.]

Коммутативное кольцо K с единицей $e_K \neq 0_K$, в котором все элементы кроме нулевого обратимы, называется **полем**. Таким образом, любое поле содержит не менее двух элементов.

Порядком поля называют количество элементов в нем.

(П) Кольцо целых чисел не образует поля, а кольцо рациональных или вещественных — образует.

(П) Кольцо классов вычетов по модулю n образует поле только в том случае, когда n — простое число, т. к. если n — составное, то в этом кольце есть делители нуля, а они не могут быть обратимы.

(П) $\mathbb{C} = (\mathbb{R} \times \mathbb{R}, +, *)$ — поле комплексных чисел, $\mathbb{C} = \{\langle a, b\rangle \mid a, b \in \mathbb{R}\}$, $\langle a, b\rangle = a + ib$, где $i = \sqrt{-1}$. Действия с комплексными числами выполняются по следующим правилам $\langle a, b\rangle + \langle c, d\rangle = \langle a + c, b + d\rangle$, $\langle a, b\rangle * \langle c, d\rangle = \langle ac - bd, ad + bc\rangle$. Единичный, нулевой и обратные элементы определяются так: $e_{\mathbb{C}} = \langle 1,\ 0\rangle$, $0_{\mathbb{C}} = \langle 0,\ 0\rangle$, $-\langle a,\ b\rangle = \langle -a,\ -b\rangle$, $\langle a, b\rangle^{-1} = \langle a/(a^2 + b^2), -b/(a^2 + b^2)\rangle$.

Характеристикой поля P называется минимальное число n такое, что $\sum_{i=1}^{n} e_P = 0_P$. Если такого n не существует, то считается, что характеристика равна 0.

(У57) Характеристика поля либо 0, либо простое число. [По определению поля характеристика $\chi \neq 1$. Предположим, что $\chi > 0$ — со-

ставное число, $\chi = ab$. Тогда

$$\underbrace{e_P + \cdots + e_P}_{\chi \text{ раз}} = 0_P$$

и, следовательно,

$$\underbrace{(e_P + \cdots + e_P)}_{a \text{ раз}} \times \underbrace{(e_P + \cdots + e_P)}_{b \text{ раз}} = 0_P.$$

В поле нет делителей нуля. Следовательно, $a = 0$ или $b = 0$ и, поэтому, $\chi = 0$. Итак, предположение о том, что $\chi = ab > 0$, — неверно.]

(П) $(\mathbb{Q}, +, *)$ — это поле характеристики 0.

(П) $(\mathbb{Z}_n, +, *)$ — это поле характеристики n (n — простое).

Кольцо многочленов

Пусть P — поле. Тогда $P[x]$ — это все возможные многочлены с коэффициентами из P.

Степенью многочлена $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \in P[x]$, где $a_n \neq 0_P$, называется число $n \in \mathbb{Z}_+$. Обозначение $\deg(f) = n$.

Нулевым называется многочлен тождественно равный нулю. Он не имеет степени.

Два многочлена называются **равными**, если равны их степени и коэффициенты при равных степенях неизвестного или оба они нулевые.

Суммой двух многочленов называется многочлен, получаемый сложением их членов и последующей группировкой коэффициентов при равных степенях неизвестного.

Произведением двух многочленов называется многочлен, получаемый перемножением их членов и последующей группировкой коэффициентов при равных степенях неизвестного.

(П) $f(x) = x^2 - 2$, $g(x) = x^3 - x^2 + 1$, $f, g \in \mathbb{Q}[x]$, $f(x) + g(x) = x^3 - 1$, $f(x)g(x) = x^5 - x^4 - 2x^3 + 3x^2 - 2$.

(У58) $\deg(f + g) \leqslant \max(\deg(f), \deg(g))$, где f и g — многочлены. [Очевидно]

(У59) $\deg(fg) = \deg(f) + \deg(g)$, где f и g — многочлены. [Очевидно]

(У60) $(P[x], +, *)$ — коммутативное кольцо с единицей и без делителей нуля. [[1, стр. 312–314]]

(У61) $\forall f, g \in P[x]$, $g \neq 0_P$ $\exists h, r \in P[x]$: $f = gh + r$, $\deg(r) < \deg(g)$ и h, r — единственны. [[3, стр. 314]]

Если $f = gh + r$ и $\deg(r) < \deg(g)$, то многочлен r называется **остатком** от деления f на g, а h — **частным**.

Говорят, что многочлен f **делится** на многочлен g, если остаток от деления f на g равен нулевому многочлену.

Многочлен h называется **наибольшим общим делителем** (НОД) многочленов f и g, если: 1) h — общий делитель f и g, т. е. f и g делятся на h; 2) h делится на любой общий делитель f и g. Обозначение $h = \text{НОД}(f, g)$.

(У62) НОД определен с точностью до постоянного множителя из P. [Очевидно]

(У63) **Алгоритм Евклида.** $\forall f,\ g \in P[x]$ следующая последовательность операций приводит к вычислению их НОД:

$$\begin{aligned}
&1)\ f = gh_1 + r_1;\\
&2)\ g = r_1h_2 + r_2;\\
&3)\ r_1 = r_2h_3 + r_3;\\
&\quad\dots\\
&k)\ r_{k-2} = r_{k-1}h_k + r_k;\\
&k+1)\ r_{k-1} = r_kh_{k+1}.
\end{aligned}$$

$\text{НОД}(f, g) = r_k$. [[3, стр. 37]]

(П) $f(x) = x^3 - 1$, $g(x) = x^2 + x - 2$. 1) $f = g * (x - 1) + (3x - 3)$; 2) $g = (3x - 3)(x/3 + 2/3) \Rightarrow \text{НОД}(f, g) = 3x - 3$ или $\text{НОД}(f, g) = x - 1$.

Многочлены f и g называются **взаимно простыми**, если $\deg(\text{НОД}(f, g)) = 0$.

$\triangledown$(У64) Для любых f и g из $P[x]$ существуют такие u и v из $P[x]$, что $fu + gv = \text{НОД}(f, g)$. [Рассмотрим последовательность операций из алгоритма Евклида 1) $f = gh_1 + r_1$; 2) $g = r_1h_2 + r_2$; 3) $r_1 = r_2h_3 + r_3$; 4) $r_2 = r_3h_4 + r_4$; … k) $r_{k-2} = r_{k-1}h_k + r_k$; k+1) $r_{k-1} = r_kh_{k+1}$. Выразим r_1 из шага 1: $r_1 = f - gh_1$. Подставим полученное значение в шаг 2: $g = (f - gh_1)h_2 + r_2$ или $r_2 = g(1 + h_1h_2) - fh_2 = gv_2 + fu_2$, где v_2 и u_2 из $P[x]$. Подставим полученные значения для r_2 и r_1 в шаг 3: $r_2 = gv_2 + fu_2 = (f - gh_1)h_3 + r_3$ или $r_3 = gv_3 + fu_3$, где v_3 и u_3 из $P[x]$. Подставим полученные значения для r_3 и r_2 в шаг 4: $r_3 = gv_3 + fu_3 = (gv_2 + fu_2)h_4 + r_4$ или $r_4 = gv_4 + fu_4$, где v_4 и u_4 из $P[x]$. Продолжая подобные операции, на k шаге получим $r_{k-2} = gv_{k-2} + fu_{k-2} = (gv_{k-1} + fu_{k-1})h_k + r_k$ или $r_k = gv + fu$, где v и u из $P[x]$.]

Многочлен h называется **наименьшим общим кратным** (НОК) многочленов f и g, если: 1) h — делится на f и g; 2) любое общее кратное f и g делится на h. Обозначение $h = \text{НОК}(f, g)$.

(У65) $\text{НОК}(f, g) = f * g/\text{НОД}(f, g)$. [$f = \text{НОД}(f, g) * \varphi$, $g = \text{НОД}(f, g) * \psi$, φ и ψ из $P[x]$ такие, что $\text{НОД}(\varphi, \psi) = 1$; следовательно, $\text{НОК}(f, g) = \varphi * \psi * \text{НОД}(f, g) = f * g/\text{НОД}(f, g)$.]

(У66) НОК определен с точностью до постоянного множителя из P. [Очевидно]

Корнем многочлена $f(x)$ из $P[x]$ называется такой элемент x_0 из P, что $f(x_0) = 0_P$.

(У67) Если x_0 — корень многочлена $f(x) \in P[x]$, то $f(x) = (x - x_0)g(x)$, где $g(x) \in P[x]$. [При делении $f(x)$ на $(x-x_0)$ получается $f(x) = (x - x_0)g(x) + r(x)$, $\deg(r) < 1$, т.е. r — константа. При подставке в f $x = x_0$ получается $f(x_0) = (x_0 - x_0)g(x) + r = 0_P$, т.е. $r = 0_P$.]

Многочлен $f \in P[x]$, $\deg(f) > 0$ называется **неприводимым** над полем P, если он делится только на себя и на многочлен нулевой степени.

Поле P называется **алгебраически замкнутым**, если неприводимыми над полем P являются лишь многочлены первой степени.

(У68) **Основная теорема алгебры.** Поле комплексных чисел алгебраически замкнуто. [[19, стр. 239–241]]

(П) Поля вещественных и рациональных чисел замкнутыми не являются. Не являются замкнутыми и поля классов вычетов по модулю n.

Алгебра логики

Функцией алгебры логики или **булевой функцией** называется отображение $f: U^n \to U$, где $U = \{0, 1\}$, а $n \in \mathbb{Z}_+$ равно числу аргументов функции.

Булевы функции — это очень узкий класс функций. Но детальное их изучение очень важно, т.к. практически все компоненты современных компьютеров могут быть описаны с их помощью.

Любую функцию алгебры логики можно задать при помощи таблицы. Для этого в левой части таблицы выписывают все возможные значения аргументов (двоичные числа от 0 до $2^n - 1$), а в правой — соответствующие им значения заданной функции:

	x_1 ... x_{n-1} x_n	$f(x_1, \ldots, x_{n-1}, x_n)$
0	0 ... 0 0	$f(0, \ldots, 0, 0)$
1	0 ... 0 1	$f(0, \ldots, 0, 1)$
2	0 ... 1 0	$f(0, \ldots, 1, 0)$
...	....	...
$2^n - 1$	1 ... 1 1	$f(1, \ldots, 1, 1)$

.

(У69) Число всех булевых функций, зависящих от n переменных, равно 2^{2^n}. [Это следует из того, что в таблице, задающей булеву функцию, ровно 2^n строк. В каждой строке в качестве значения функции можно взять либо 0, либо 1. Из этого и следует утверждение.]

(П) Из этого утверждения следует, что число функций алгебры логики конечно, но оно очень быстро растет с ростом n: при $n = 1$ это число равно 4, при $n = 2$ — 16, при $n = 3$ — 256, при $n = 4$ — 65536... При $n > 5$ перебор всех функций становится практически невозможным даже при использовании вычислительной техники.

Булева функция $f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ **зависит существенным образом** от аргумента x_i, если существуют такие значения $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$ переменных $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, что $f(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n) \neq f(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n)$. В этом случае переменная x_i называется **существенной**. Если x_i не является существенной переменной, то она называется **несущественной** или **фиктивной**.

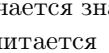
Пусть для $f(x_1, \ldots, x_n)$ переменная x_i $(1 \leqslant x_i \leqslant n)$ является фиктивной. Из таблицы для функции f путем вычеркивания всех строк вида $\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n$ и вычеркиванием столбца для аргумента x_i получается новая таблица. Полученная таблица будет определять некоторую функцию $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Функция g получена из f путем **удаления фиктивной переменной** x_i, а функция f получается из g путем **введения фиктивной переменной** x_i.

Две булевы функции f и g называются **равными**, если функцию f можно получить из g путем добавления или изъятия фиктивных переменных.

Существуют два типа функций, не имеющих существенных переменных: функции первого типа равны тождественно 0, второго — 1. Эти функции-константы будем обозначаются 0 и 1.

Если задана конечная система булевых функций $\{f_1, \ldots, f_s\}$, где $s > 0$, то можно считать, что все эти функции зависят от одних и тех же переменных.

Рассмотрим наиболее часто используемые функции алгебры логики. В математической логике и кибернетике они играют роль подобную степенным и тригонометрическим функциям в математическом анализе, поэтому их можно считать "элементарными" функциями:

1) константа 0;
2) константа 1;
3) $f(x) = x$, тождественная функция;
4) $f(x) = \neg x$, отрицание x ($\neg x$ читается "не x"). Часто вместо знака $\neg$ над переменной x ставят черту. В большинстве языков программирования для записи отрицания в выражениях используется функция NOT. На электронных схемах обозначается знаком  ;
5) $f(x_1, x_2) = (x_1 \,\&\, x_2)$, конъюнкция x_1 и x_2 (читается "x_1 и x_2"), $(x_1 \,\&\, x_2) = \min(x_1, x_2)$. Вместо знака & иногда используется знак

$\wedge$ или вообще знак опускается, т. е. пишут (x_1x_2). Эту функцию часто называют логическим умножением — она совпадает с арифметическим умножением. В большинстве языков программирования для записи конъюнкции в выражениях используется операция AND. На электронных схемах обозначается знаком  ;

6) $f(x_1, x_2) = (x_1 \vee x_2)$, дизъюнкция x_1 и x_2 (читается "x_1 или x_2"), $(x_1 \vee x_2) = \max(x_1, x_2)$. Эту функцию иногда называют логическим сложением. В большинстве языков программирования для записи дизъюнкции в выражениях используется операция OR. На электронных схемах обозначается знаком — ;
7) $f(x_1, x_2) = (x_1 \Rightarrow x_2)$, импликация x_1 и x_2 (читается "из x_1 следует x_2"). Эту функцию часто называют логическим следованием. В языке программирования Бэйсик для записи импликации в выражениях используется операция IMP;
8) $f(x_1, x_2) = (x_1 \sim x_2)$, эквивалентность x_1 и x_2 (читается "x_1 эквивалентен x_2"). В Бэйсике для записи эквивалентности в выражениях используется операция EQV. На электронных схемах обозначается знаком ;
9) $f(x_1, x_2) = (x_1 + x_2)$, сложение по модулю 2 x_1 и x_2. Эту функцию часто называют "исключающим ИЛИ". В большинстве языков программирования для записи этой операции используется слово XOR. На электронных схемах обозначается знаком ;
10) $f(x_1,\ x_2) = (x_1 \mid x_2)$, функция Шеффера или штрих Шеффера, антиконъюнкция, элемент "И-НЕ" электронных схем ();
11) $f(x_1,\ x_2) = (x_1 \downarrow x_2)$, функция Пирса или стрелка Пирса, антидизъюнкция, элемент "ИЛИ-НЕ" электронных схем ().

Таблицы для этих функций имеют следующий вид:

x	$f(x)$			
	0	1	x	$\neg x$
0	0	1	0	1
1	0	1	1	0

$x_1\,x_2$	$f(x_1, x_2)$						
	$(x_1\ \&\ x_2)$	$(x_1 \vee x_2)$	$(x_1 \Rightarrow x_2)$	$(x_1 \sim x_2)$	$(x_1 + x_2)$	$(x_1 \mid x_2)$	$(x_1 \downarrow x_2)$
0 0	0	0	1	1	0	1	1
0 1	0	1	1	0	1	1	0
1 0	0	1	0	0	1	1	0
1 1	1	1	1	1	0	0	0

.

Формулы. Реализация булевых функций формулами

Пусть B некоторое множество булевых функций, $B = \{f_1, f_2, \ldots,$

$f_s, \ldots\}$. Тогда:

1) Каждая функция $f_i(x_1, \ldots, x_{n_i}) \in B$ называется **формулой** над B;
2) Пусть $f_i(x_1, \ldots, x_{n_i}) \in B$ и A_1, …, A_{n_i} — выражения, являющиеся либо формулами над B, либо символами переменных. Тогда выражение $f_i(A_1, \ldots, A_{n_i})$ называется **формулой** над B (рекурсивное определение).

(П) Пусть B — множество элементарных булевых функций. Следующие выражения являются формулами над B: 1) $(((x_1x_2)+x_1)\vee x_2)$; 2) $(\neg x_1(x_1+x_2))$. Выражение $+x_1x_2$ формулой над B не является.

Если функция f соответствует формуле V, то говорят, что V **реализует** f.

Функция f, соответствующая формуле V, называется **суперпозицией** функций из B, а процесс получения функции f из B — **операцией суперпозиции**.

Формулы V и W называются **эквивалентными**, если соответствующие им функции f_V и f_W равны, т.е. $f_V = f_W$. Запись $W = V$ будет означать, что W и V эквивалентны.

(П) 0 эквивалентна $(\neg x \;\&\; x)$, $(x \Rightarrow y) = (\neg x \vee y) = (\neg y \Rightarrow \neg x)$, $\neg(x+y) = (x \sim y)$.

(У70) Пусть $B = \{0, 1, x, \neg x, (x \;\&\; y), (x \vee y)\}$. Тогда имеют место следующие свойства:

1) $(x \;\&\; x) = x$, $(x \vee x) = x$ (идемпотентность);
2) $(x \;\&\; y) = (y \;\&\; x)$, $(x \vee y) = (y \vee x)$ (коммутативность);
3) $((x \;\&\; y) \;\&\; z) = (x \;\&\; (y \;\&\; z))$, $((x \vee y) \vee z) = (x \vee (y \vee z))$ (ассоциативность);
4) $(x \;\&\; (x \vee y)) = (x \vee (x \;\&\; y)) = x$ (законы поглощения);
5) из $x \leqslant z$ следует $(x \vee (y \;\&\; z)) = ((x \vee y) \;\&\; z)$ (модулярный закон);
6) $(x \;\&\; (y \vee z)) = ((x \;\&\; y) \vee (x \;\&\; z))$, $(x \vee (y \;\&\; z)) = ((x \vee y) \;\&\; (x \vee z))$ (дистрибутивность);
7) $(x \;\&\; 0) = 0$, $(x \vee 0) = x$, $(x \;\&\; 1) = x$, $(x \vee 1) = 1$ (универсальные границы);
8) $(x \;\&\; \neg x) = 0$, $(x \vee \neg x) = 1$ (дополняемость);
9) $\neg\neg x = x$ (инволютивность);
10) $\neg(x \;\&\; y) = (\neg x \vee \neg y)$, $\neg(x \vee y) = (\neg x \;\&\; \neg y)$ (законы де Моргана). [Проверка при помощи таблиц функций.]

(У71) Множество $\{0, 1\}$ относительно операций отрицания, конъюнкции и дизъюнкции есть булева алгебра. [Следует из предыдущего утверждения и определения булевой алгебры.]

С целью упрощения записи формул можно условиться, что: 1) операция & сильнее операции $\vee$, т.е. если нет скобок, то сначала выполняется операция &, а потом $\vee$, например, запись $(x \;\&\; y \vee z)$ означает

$((x \,\&\, y) \vee z)$; 2) в силу закона ассоциативности для операций $\&$, $\vee$ и $+$ можно вместо формул $((x \circ y) \circ z)$, $(x \circ (y \circ z))$ пользоваться выражением $(x \circ y \circ z)$, где $\circ \in \{\&, \vee, +\}$; 3) внешние скобки можно опускать. Полученные в результате такого упрощения выражения не будут формулами, но они однозначно могут быть превращены в формулы путем расстановки скобок.

Обозначения: 1) логическое произведение — $\&_{i=1}^{s} x_i = x_1 \,\&\, x_2 \,\&\, \cdots \,\&\, x_s$; 2) логическая сумма — $\bigvee_{i=1}^{s} x_i = x_1 \vee x_2 \vee \cdots \vee x_s$.

(У72) Если в логическом произведении один из множителей равен 0, то и все произведение равно 0. Если в логическом произведении один из множителей равен 1, то этот множитель можно зачеркнуть. Если в логической сумме одно слагаемое равно 0, то его можно зачеркнуть. Если в логической сумме одно из слагаемых равно 1, то и вся сумма равна 1. [Очевидно]

Функция $\neg f(\neg x_1, \ldots, \neg x_n)$ называется **двойственной функцией** к функции $f(x_1, \ldots, x_n)$ и обозначается $f^*(x_1, \ldots, x_n)$.

(У73) Таблица для двойственной функции f^* получается из таблицы для исходной функции f инвертированием (заменой 0 на 1 и 1 на 0) столбца значений функции и его переворачиванием. [Очевидно]

(П) 0 двойственна к 1, 1 двойственна к 0, x двойственна к x, $\neg x$ двойственна к $\neg x$, $x \,\&\, y$ двойственна к $x \vee y$, $x \vee y$ двойственна к $x \,\&\, y$.

(У74) $f^{**} = f$. [Очевидно]

(У75) **Принцип двойственности.** Если формула V над $B = \{f_1, f_2, \ldots, f_s\}$ реализует функцию f, то формула V^* над $B^* = \{f_1^*, f_2^*, \ldots, f_s^*\}$, полученная из V заменой всех f_i на f_i^* $(1 \leqslant i \leqslant s)$ реализует функцию f^*. [Очевидно]

(У76) Пусть V и W — формулы над B. Тогда из $V = W$ следует, что $V^* = W^*$, где V^* и W^* — формулы над B^*. [Следует из принципа двойственности.]

Далее занимаемся решением следующих важных задач: 1) Если в качестве B взять некоторые функции алгебры логики, то всякая ли булева функция сможет быть выражена в виде формулы над B? 2) Какие функции нужно включить в B, чтобы любая функция алгебры логики выражалась в виде формулы над B?

Введем обозначение $x^\sigma = x\sigma \vee \neg x\neg\sigma$, где $\sigma \in \{0, 1\}$, т.е.

$$x^\sigma = \begin{cases} \neg x, & \text{при } \sigma = 0; \\ x, & \text{при } \sigma = 1. \end{cases}$$

Следовательно, $x^\sigma = 1$ тогда и только тогда, когда $x = \sigma$.

(У77) **Разложение булевой функции по переменным.** Каждую булеву функцию $f(x_1, \ldots, x_n)$ при любом m $(1 \leqslant m \leqslant n)$ можно

представить в форме $f(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) = \bigvee_{(\sigma_1,\ldots,\sigma_m)} x_1^{\sigma_1}$ & $\cdots$ & $x_m^{\sigma_m}$ & $f(\sigma_1, \ldots, \sigma_m, x_{m+1}, \ldots, x_n)$, где дизъюнкция берется по всевозможным наборам значений переменных $x_1, \ldots, x_m$. Эта форма называется разложением функции по m переменным $x_1, \ldots, x_m$. [Рассмотрим произвольный набор значений переменных $\langle\alpha_1, \ldots, \alpha_n\rangle$ и покажем, что левая и правая части доказываемого соотношения принимают на нем одно и то же значение. Левая часть дает $f(\alpha_1, \ldots, \alpha_n)$. Правая —

$$\bigvee_{\langle\sigma_1,\ldots,\sigma_m\rangle} \alpha_1^{\sigma_1} \,\&\, \cdots \,\&\, \alpha_m^{\sigma_m} \,\&\, f(\sigma_1, \ldots, \sigma_m, \alpha_{m+1}, \ldots, \alpha_n) =$$

$$= \alpha_1^{\alpha_1} \,\&\, \cdots \,\&\, \alpha_m^{\alpha^m} \,\&\, f(\alpha_1, \ldots, \alpha_m, \alpha_{m+1}, \ldots, \alpha_n) =$$

$= f(\alpha_1, \ldots, \alpha_n)$.]

Разложение

$$f(x_1, x_2, \ldots, x_n) = \neg x_1 \,\&\, f(0, x_2, \ldots, x_m) \vee x_1 \,\&\, f(1, x_2, \ldots, x_m)$$

называется **разложением по переменной** x_1. Функции

$$f(0, x_2, \ldots, x_m) \text{ и } f(1, x_2, \ldots, x_m)$$

называются **компонентами разложения**.

Разложение

$$f(x_1, \ldots, x_n) = \bigvee_{\langle\sigma_1,\ldots,\sigma_n\rangle} x_1^{\sigma_1} \,\&\, \cdots \,\&\, x_n^{\sigma_n} \,\&\, f(\sigma_1, \ldots, \sigma_n)$$

называется **разложением по всем переменным**.

Если $f(x_1, \ldots, x_n)$ не есть нулевая функция, то правая часть формулы

$$f(x_1, \ldots, x_n) = \bigvee_{\langle\sigma_1,\ldots,\sigma_n\rangle, f(\sigma_1,\ldots,\sigma_n)=1} x_1^{\sigma_1} \,\&\, \cdots \,\&\, x_n^{\sigma_n}$$

называется **совершенной дизъюнктивной нормальной формой** (СДНФ).

(У78) Каждая функция алгебры логики может быть выражена в виде формулы через отрицание, конъюнкцию и дизъюнкцию. [1) Пусть $f(x_1, \ldots, x_n)$ — нулевая. Тогда $f(x_1, \ldots, x_n) = x_1 \,\&\, \neg x_1$. 2) Пусть $f(x_1, \ldots, x_n)$ не есть нулевая. Тогда представим ее в виде СДНФ.]

Доказано, что любую функцию алгебры логики можно задать формулой над $B = \{\neg x, x \,\&\, y, x \vee y\}$. Доказательство проведено конструктивно, т. е. в нем показано как для любой функции алгебры логики

$f(x_1, \ldots, x_n)$ построить формулу ее реализующую и содержащую только конъюнкцию, дизъюнкцию и отрицание: 1) в таблице, задающей ненулевую $f(x_1, \ldots, x_n)$ отмечаются все строки аргументов $(\sigma_1, \ldots, \sigma_n)$ таких, что $f(\sigma_1, \ldots, \sigma_n) = 1$; 2) для каждой такой строки образуем логическое произведение; 3) все полученные конъюнкции соединяем знаком дизъюнкции.

(П) Напишем СДНФ для функции $f(x, y) = x \Rightarrow y$. Имеется три набора, на которых эта функция равна 1: (0,0), (0,1) и (1,1). Поэтому $f(x, y) = \neg x \neg y \vee \neg x y \vee x y$.

(У79) Если $f(x_1, \ldots, x_n)$ не есть единичная функция, то

$$f(x_1, \ldots, x_n) = \&_{\langle \sigma_1, \ldots, \sigma_n \rangle, f(\sigma_1, \ldots, \sigma_n) = 0} \, (x_1^{\neg \sigma_1} \vee \cdots \vee x_n^{\neg \sigma_n}).$$

Правая часть этого разложения называется **совершенной конъюнктивной нормальной формой** (СКНФ). [Если f — не единичная, то f^* — не нулевая. Следовательно, f^* можно представить в виде СДНФ:

$$f^*(x_1, \ldots, x_n) = \bigvee_{\langle \sigma_1, \ldots, \sigma_n \rangle, f^*(\sigma_1, \ldots, \sigma_n) = 1} x_1^{\sigma_1} \,\&\, \cdots \,\&\, x_n^{\sigma_n}.$$

Известно, что $f^{**} = f$. Следовательно, $f = (f^*)^* =$

$$= (\bigvee_{\langle \sigma_1, \ldots, \sigma_n \rangle, f^*(\sigma_1, \ldots, \sigma_n) = 1} x_1^{\sigma_1} \,\&\, \cdots \,\&\, x_n^{\sigma_n})^* =$$

$$= \neg (\bigvee_{\langle \sigma_1, \ldots, \sigma_n \rangle, f^*(\sigma_1, \ldots, \sigma_n) = 1} x_1^{\neg \sigma_1} \,\&\, \cdots \,\&\, x_n^{\neg \sigma_n}) =$$

$$= \&_{\langle \sigma_1, \ldots, \sigma_n \rangle, f(\neg \sigma_1, \ldots, \neg \sigma_n) = 0} \, (x_1^{\sigma_1} \vee \cdots \vee x_n^{\sigma_n}) =$$

$$= \&_{\langle \sigma_1, \ldots, \sigma_n \rangle, f(\sigma_1, \ldots, \sigma_n) = 0} \, (x_1^{\neg \sigma_1} \vee \cdots \vee x_n^{\neg \sigma_n}).]$$

(П) Построить СКНФ для функции $f(x, y) = x \Rightarrow y$. Имеется один набор, на котором эта функция равна 0 — (1,0). Поэтому $f(x, y) = \neg x \vee y$.

Формула, состоящая из дизъюнкции конъюнкций символов переменных или их отрицаний, называется **дизъюнктивной нормальной формой** — ДНФ, а формула, состоящая из конъюнкции дизъюнкций символов переменных или их отрицаний, называется **конъюнктивной нормальной формой** — КНФ.

(П) ДНФ для $x \Rightarrow y$ будут, например, $\neg x \neg y \vee \neg x y \vee x y$ (СДНФ), $\neg x \vee x y$, $\neg x \vee y$.

ДНФ для заданной функции, содержащую минимум символов переменных, называют **минимальной**.

Задача построения минимальной ДНФ неперебо́рными методами является актуальной для цифровой электроники и до сих пор не имеет оптимального решения.

(П) Любую булеву функцию можно задать не только таблицей, но и формулой, содержащей только конъюнкцию, дизъюнкцию и отрицание. Как правило, язык таких формул гораздо удобнее, чем таблицы. Например, формула для задания функции $f(x_1, \ldots, x_{20}) = x_1 \,\&\, \cdots \,\&\, x_{20}$ содержит всего 39 символов (20 символов переменных и 19 знаков &). Таблица, задающая эту же функцию, содержит 2^{20}, т. е. более миллиона строк.

Система функций $\{f_1, \ldots, f_s, \ldots\}$ называется (**функционально**) **полной**, если любая булева функция может быть записана в виде формулы через функции этой системы.

(П) Полными системами являются: 1) все функции алгебры логики; 2) система $\{\neg x, x \vee y, x \,\&\, y\}$. Система $\{0, 1\}$ не является полной.

(У80) Пусть даны две системы булевых функций: $B = \{f_1, f_2, \ldots\}$ и $C = \{g_1, g_2, \ldots\}$. Известно, что система B полна и каждая ее функция выражается в виде формулы над C. Тогда система C является полной. [Следует из определения формулы.]

(П) Система $C = \{\neg x, x \vee y\}$ — полна. Это следует из того, что $B = \{\neg x, x \vee y, x \,\&\, y\}$ — полна и $x \vee y = \neg(\neg x \,\&\, \neg y)$.

(П) Система $C = \{\neg x, x \,\&\, y\}$ — полна. Это следует из того, что $B = \{\neg x, x \vee y, x \,\&\, y\}$ — полна и $x \,\&\, y = \neg(\neg x \vee \neg y)$. Доказательство также можно провести, используя принцип двойственности.

(П) Система $C = \{x \mid y\}$ — полна. Это следует из того, что $B = \{\neg x, x \,\&\, y\}$ — полна, $\neg x = x \mid x$ и $x \,\&\, y = (x \mid y) \mid (x \mid y)$.

(П) Система $C = \{1, x \,\&\, y, x + y\}$ — полна. Это следует из того, что $B = \{\neg x, x \,\&\, y\}$ — полна и $\neg x = x + 1$.

Каждая формула над $C = \{1, xy, x + y\}$ после раскрытия скобок и несложных алгебраических преобразований переходит в полином, называемый **полиномом Жегалкина**.

(П) Построить полином Жегалкина для функций $f(x, y, z) = x \,\&\, \neg y \,\&\, z$ и $g(x, y) = x \vee y$. Получаем $f(x, y, z) = x(y + 1)z = xyz + xz$, $g(x, y) = \neg(\neg x \,\&\, \neg y) = ((x + 1)(y + 1)) + 1 = xy + x + y$. Вид формулы для g можно искать, используя метод неопределенных коэффициентов: $g(x, y) = axy + bx + cy + d$. При $x = y = 0$ получаем $d = 0$. При $x = 0$, $y = 1$ получаем $c = 1$. При $x = 1$, $y = 0$ получаем $b = 1$. При $x = y = 1$ получаем $a = 1$.

(У81) Каждая функция алгебры логики выражается в виде полинома Жегалкина единственным образом. [Каждое слагаемое полинома Жегалкина для булевой функции от n переменных имеет вид произ-

ведения всех элементов некоторого подмножества множества $U = \{x_1, \ldots, x_n\}$ на некоторый коэффициент. Количество таких подмножеств и, следовательно, слагаемых равно 2^n, т. к. $\#U = n$ и $\#P(U) = 2^n$. В качестве коэффициента при каждом из слагаемых можно взять либо 0, либо 1. Таким образом, всего может быть ровно 2^{2^n} разных полиномов Жегалкина от n переменных или столько же сколько и разных булевых функций от n переменных.]

Пусть B некоторое множество булевых функций. **Замыканием** B называется множество всех булевых функций, представимых в виде формул над B. Замыкание множества B обозначается через $[B]$.

(П) B — все булевы функции. Очевидно, что $[B] = B$.

(П) $B = \{1, x + y\}$. Замыканием B будет класс L всех линейных функций, т. е. функций, имеющих вид $f(x_1, \ldots, x_n) = c_0 + c_1x_1 + \cdots + c_nx_n$, где $c_i \in \{0, 1\}$. Функции 0, 1, x, $\neg x$, $x + y$ входят в L, а функции $x \,\&\, y$ и $x \vee y$ не входят.

(У82) Свойства замыкания: 1) $B \subset [B]$; 2) $[[B]] = [B]$; 3) $B \subset C \Rightarrow [B] \subset [C]$; 4) $[B] \cup [C] \subset [B \cup C]$. [Очевидно]

Класс (множество) B называется (функционально) **замкнутым**, если $[B] = B$.

(П) Класс всех булевых функций замкнут. Класс $B = \{1, x + y\}$ не замкнут.

(У83) Класс L замкнут. [Линейное выражение, составленное из линейных выражений, является линейным.]

(Определение полноты через понятие замыкания) Система функций называется **полной**, если ее замыкание — это множество всех булевых функций.

Важнейшие замкнутые классы

Обозначим через T_0 класс всех булевых функций, сохраняющих константу 0, т. е. таких, что $f(0, \ldots, 0) = 0$. Функции 0, x, $x \,\&\, y$, $x \vee y$ входят в T_0, а функции 1 и $\neg x$ — не входят.

(У84) Количество всех функций от n переменных из T_0 равно $2^{2^n}/2 = 2^{2^n-1}$. [Таблица любой функции из T_0 на нулевой строке аргументов принимает нулевое значение, а в остальном произвольна.]

Обозначим через T_1 класс всех булевых функций, сохраняющих константу 1, т. е. таких, что $f(1, \ldots, 1) = 1$. Функции 1, x, $x \,\&\, y$, $x \vee y$ входят в T_1, а функции 0 и $\neg x$ — не входят.

(У85) Классы T_0 и T_1 — замкнуты. [[13, стр. 52]]

(У86) Количество всех функций от n переменных из T_1 равно $2^{2^n}/2 = 2^{2^n-1}$. [Аналогично случаю T_0.]

Обозначим через S класс всех булевых самодвойственных функций, т. е. таких, что $f = f^*$. Функции $\neg x, x, xy \vee xz \vee yz$ входят в S, а функции

0, 1, $x \,\&\, y$ и $x \vee y$ — не входят.

(У87) Класс S — замкнут. [[13, стр. 53]]

(У88) Количество всех функций от n переменных из S равно $\sqrt{2^{2^n}} = 2^{2^{n-1}}$. [Для самодвойственной функции верно, что

$$f(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n).$$

Следовательно, самодвойственная функция полностью определяется на половине множества значений своих аргументов, т. е. на 2^{n-1} строках таблицы.]

(У89) Количество всех функций от n переменных из L равно 2^{n+1}. [Любая линейная функция от n переменных задается значением $(n+1)$ коэффициента, каждый из которых либо 0, либо 1.]

Считается, что набор $\langle \alpha_1, \ldots, \alpha_n \rangle$ **не больше** набора $\langle \beta_1, \ldots, \beta_n \rangle$, если для любого i $(1 \leqslant i \leqslant n)$ $\alpha_i \leqslant \beta_i$.

(П) $\langle 0, 1, 0, 1 \rangle \leqslant \langle 1, 1, 0, 1 \rangle$, но неверно, что $\langle 1, 0 \rangle \leqslant \langle 0, 1 \rangle$.

Булева функция $f(x_1, \ldots, x_n)$ называется **монотонной**, если для любых наборов $\langle \alpha_1, \ldots, \alpha_n \rangle \leqslant \langle \beta_1, \ldots, \beta_n \rangle$ имеет место неравенство $f(\alpha_1, \ldots, \alpha_n) \leqslant f(\beta_1, \ldots, \beta_n)$. Обозначим через M класс всех монотонных функций.

(П) Монотонными являются функции 0, 1, x, $x \,\&\, y$, $x \vee y$. Функции $\neg x$ и $x \mid y$ монотонными не являются.

(У90) Класс M — замкнут. [[13, стр. 53–54]]

(У91) Замкнутые классы T_0, T_1, S, M и L попарно различны. [Рассмотрим таблицу:

	T_0	T_1	S	M	L
0	$\in$	$\notin$	$\notin$	$\in$	$\in$
1	$\notin$	$\in$	$\notin$	$\in$	$\in$
$\neg x$	$\notin$	$\notin$	$\in$	$\notin$	$\in$

.]

(У92) **Критерий Поста.** Для того, чтобы система функций B была полной, необходимо и достаточно, чтобы она целиком не содержалась ни в одном из пяти замкнутых классов T_0, T_1, S, M и L. [[20, стр. 32–33]]

(У93) Всякий замкнутый класс булевых функций, не совпадающий с множеством всех булевых функций, содержится по крайней мере в одном из классов T_0, T_1, S, M или L. [Следует из предыдущей теоремы.]

Класс булевых функций B называется **предполным** (или **максимальным**), если B неполный, а для любой булевой функции $f \notin B$ класс $B \cup \{f\}$ — полный.

(У94) Любой предполный класс является замкнутым. [Пусть B — предполный незамкнутый класс, $B \subset [B]$ и $B \neq [B]$. Тогда $\exists f$, $f \in [B]$ и $f \notin B$. Получаем $[B] = [B \cup \{f\}]$, что противоречит тому, что B — предполный (не полный) класс.]

(У95) В алгебре логики существует только пять предполных классов: T_0, T_1, S, M и L. [Пусть существует еще один предполный класс, Z. Тогда Z либо подмножество одного из классов T_0, T_1, S, M или L, либо Z не является подмножеством ни одного из них. Во втором случае Z согласно критерию Поста — полный. В первом случае, если Z не совпадает с одним из пяти известных предполных классов, то он — не предполный.]

(П) Покажем, что система функций $B = \{0, 1, xy, x+y+z\}$ является полной. Действительно $0 \notin S$, $xy \notin L$, $0 \notin T_1$, $1 \notin T_0$, $x + y + z \notin M$. Любая подсистема из трех функций из B не будет являться полной. Действительно, $\{0, 1, xy\} \subset M$, $\{1, xy, x+y+z\} \subset T_1$, $\{0, xy, x+y+z\} \subset T_0$, $\{0, 1, x + y + z\} \subset L$.

(У96) Из всякой полной системы булевых функций можно выделить полную подсистему, содержащую не более четырех функций. [[20, стр. 33]]

Графы

Набором (неупорядоченным) элементов называется неупорядоченная совокупность объектов, среди которых могут быть повторяющиеся. К наборам применима терминология, используемая при работе с множествами.

Множество $M = \{a_1, a_2, \ldots\}$ и набор N неупорядоченных пар объектов $\{a_i, a_j\}$ из M называются (неориентированным) **графом** $\Gamma = (M, N)$. Объекты множества M называются **вершинами** графа, а объекты набора N — **ребрами** графа. Про ребро $\{a_i, a_j\}$ будем говорить, что оно **соединяет** вершины a_i и a_j. Если набор N состоит из упорядоченных пар объектов $\langle a_i, a_j\rangle$ из M, то граф Γ называется **ориентированным** или **орграфом**. Ребра орграфа называют также **дугами**.

(П*) Пусть $M = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$,

$$N = \{\{a_1, a_2\}, \{a_2, a_2\}, \{a_4, a_5\}, \{a_5, a_6\}, \{a_5, a_6\}, \{a_6, a_7\}, \{a_5, a_7\}\}.$$

Тогда M и N определяют граф.

В случае, если множество M и набор N состоят из конечного числа элементов, то соответствующий им граф называется **конечным**.

Матрицей смежности конечного графа Γ называется симметричная, квадратная матрица размерности $\#M \times \#M$, получаемая следующим образом: 1) все вершины из M нумеруются; 2) в строку i столбца j ставится количество ребер, соединяющих вершины с номерами i и j. Для орграфа в строку i столбца j ставится количество ребер вида $\langle i, j\rangle$.

(П) Граф из П* однозначно определяется следующей матрицей смежности:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

(П**) Орграф Γ из $M = \{1, 2, 3, 4\}$ и $N = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 2, 3\rangle, \langle 2, 4\rangle, \langle 3, 1\rangle, \langle 4, 2\rangle\}$ однозначно определяется следующей матрицей смежности:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Система ребер графа Γ $A_{a_i,a_j} = \{\{a_i, a_{k_1}\}, \{a_{k_1}, a_{k_2}\}, \ldots, \{a_{k_s}, a_j\}\}$ называется **путем** или **маршрутом, соединяющим вершины** a_i и a_j. Для любого ребра, принадлежащего пути A_{a_i,a_j}, будем говорить, что путь A_{a_i,a_j} **проходит через** это **ребро**. Если вершина a принадлежит некоторому ребру пути A_{a_i,a_j}, то говорим, что путь A_{a_i,a_j} **проходит через вершину** a.

Путь A_{a_i,a_j} называется **циклом**, если $a_i = a_j$. Цикл $\{\{a_i, a_i\}\}$ называется **петлей**.

Цикл графа называется **гамильтоновым**, если он проходит через все вершины только один раз.

Цикл графа называется **эйлеровым**, если он проходит через все ребра только один раз.

Понятия **пути**, **цикла** и **петли** для орграфа определяются аналогично.

Граф Γ называется **связным**, если для любых двух различных его вершин существует путь, соединяющий эти вершины.

(У97) В связном графе $\Gamma = (M, N)$ $\#N \geqslant \#M - 1$. [Очевидно]

Конечный граф Γ, состоящий из m вершин $\{a_1, \ldots, a_m\}$ и C_m^2 различных ребер вида $\{a_i, a_j\}$, $1 \leqslant i < j \leqslant m$, называется **полным**. В полном графе каждые две различные вершины соединены ровно одним ребром.

(П) Граф из П* является конечным, несвязным, содержащим петли и неполным.

Если вершина a принадлежит ребру u, то a и u **инцидентны**.

Вершины a и b — **смежные** в графе Г, если они в нем определяют ребро. Ребра u и v — **смежные** в Г, если у них есть общая вершина.

Степенью вершины a графа Г называется число $\delta(a)$ ребер Г, инцидентных вершине a. Вершина, имеющая степень 0, называется **изолированной**, а степень 1 — **висящей**.

(У98) Чтобы в связном графе существовал эйлеров цикл необходимо и достаточно, чтобы в нем не было вершин нечетной степени. [[10, стр. 198]]

Матрицей инцидентности конечного графа Г называется матрица размерности $\#M \times \#N$, получаемая следующим образом: 1) все вершины из M и ребра из N нумеруются; 2) в строку i столбца j ставится либо 1, если вершина с номером i инцидентна ребру с номером j, либо 0, в противном случае. Для орграфа в строке i столбца j содержится либо 1, если ребро с номером j заканчивается вершиной с номером i, либо -1, если ребро с номером j начинается вершиной с номером i, либо 0, если вершина с номером i неинцидентна ребру с номером j.

(П) Граф из П* однозначно определяется следующей матрицей инцидентности:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

(П) Орграф из П** однозначно определяется следующей матрицей инцидентности:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & -1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}.$$

Граф $\Gamma = (M, N)$ называется **нагруженным**, если на наборе N определена функция $l\colon N \to \mathbb{R}$, называемая **весовой**. Значение $l(v)$ называется **длиной** или **весом** ребра v.

Графы Γ_1 и Γ_2 называются **изоморфными**, если существует взаимно однозначное соответствие между их вершинами и ребрами такое, что соответствующие ребра соединяют соответствующие вершины.

$\triangledown$(У99) Число попарно неизоморфных графов без изолированных вершин с h ребрами меньше, чем $e(2eh)^h$. [[20, стр. 149]]

Деревом называется связный граф без циклов, с выделенной вершиной, называемой **корнем**. Ребра дерева называют **ветвями**, висящие вершины — **листьями**.

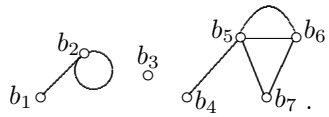
(У100) В дереве $\Gamma = (M, N)$ верно, что $\#N = \#M - 1$. [Очевидно]

(У101) Число попарно неизоморфных деревьев с h ветвями меньше, чем 4^h. [[20, стр. 154]]

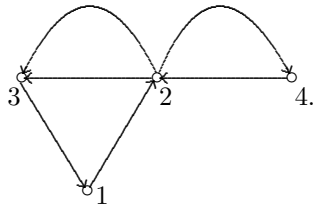
Лесом называется граф без циклов.

Фигура Ф называется **геометрической реализацией** графа Г, если существует взаимно-однозначное соответствие между вершинами фигуры Ф и вершинами графа Г, а также между линиями фигуры Ф и ребрами графа Г такое, что если линии $\{b_i, b_j\}$ соответствует ребро $\{a_i, a_j\}$, то вершине b_i соответствует вершина a_i и вершине b_j — вершина a_j. В геометрической реализации орграфа линии должны быть направленными.

(П) Графу из П* соответствует следующая геометрическая реализация:

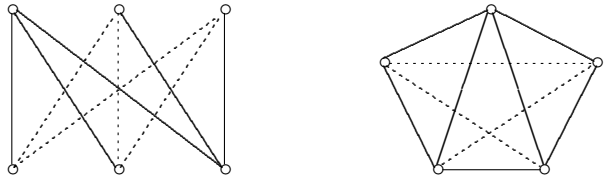


(П) Орграфу из П** соответствует следующая геометрическая реализация:



(У102) Каждый конечный граф можно реализовать в трехмерном евклидовом пространстве. [[20, стр. 145]]

(П***) Следующие два графа не допускают реализации на плоскости, т.е. в двухмерном евклидовом пространстве.



— первый из них связан с классической задачей о трех домах и трех колодцах, второй — это полный граф с пятью вершинами.

(У103) Граф и его геометрическая реализация — изоморфны. [Оче-

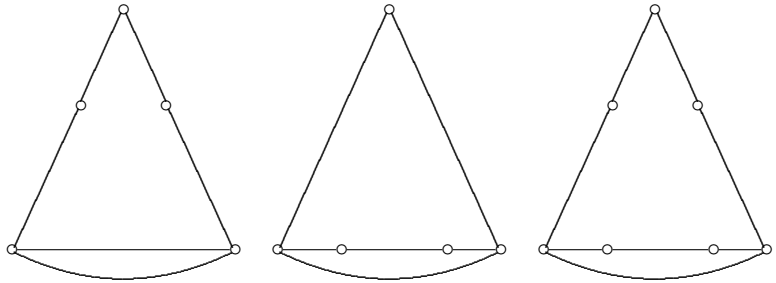
видно]

Пусть $\{a_i, a_j\}$ — произвольное ребро графа Γ и a — объект, не принадлежащий M. **Операция подразделения ребра** $\{a_i, a_j\}$ графа Γ состоит в построении графа Γ', имеющего своими вершинами множество $M' = M \cup \{a\}$, содержащего все ребра графа Γ, кроме выделенного $\{a_i, a_j\}$, и плюс два новых ребра $\{a_i, a\}$ и $\{a, a_j\}$, т.е. $N' = (N \backslash \{\{a_i, a_j\}\}) \cup \{\{a_i, a\}, \{a, a_j\}\}$.

Граф Γ_2 называется **подразделением** графа Γ_1, если он может быть получен из Γ_1 путем применения конечного числа раз операции подразделения ребер.

Графы Γ_1 и Γ_2 называются **гомеоморфными**, если существуют их изоморфные подразделения.

(П) Графы Γ_1 и Γ_2 на следующем рисунке не изоморфны, но гомеоморфны. Γ является подразделением как графа Γ_1, так и графа Γ_2.



Граф Γ' называется **подграфом** Γ, если его вершины и ребра принадлежат Γ.

(У104). **Критерий плоской реализуемости.** Для того чтобы конечный граф имел плоскую реализацию необходимо и достаточно, чтобы любой его подграф не был гомеоморфен ни одному из графов из П***. []

Алгоритм поиска кратчайшего пути на графе

Рассмотрим нагруженный граф $\Gamma = (M, N)$, $M = \{1, \ldots, n\}$, с неотрицательной весовой функцией l. Нужно найти кратчайшее расстояние между вершинами 1 и k, $1 < k \leqslant n$.

Множество вершин можно разбить на два дополнительных подмножества S и S' таких, что:

1) для каждой вершины из S' известно минимальное расстояние до вершины 1;
2) для каждой вершины из S это расстояние неизвестно.

Вначале S' состоит из одной вершины 1. Переход некоторой вершины a из подмножества S в S' произойдет тогда, когда будет доказано, что никакой путь от 1 до a через другие вершины из S не является короче, чем через вершины из S'.

Обозначим через $D'(i)$ кратчайшее расстояние между вершинами 1 и i, $i \in S'$, а через $D(i)$ кратчайшее известное между вершинами 1 и i, $i \in M$. Множества S и S' характеризуются следующими свойствами:

1) если $i \in S'$, то $D(i) = D'(i)$;
2) если $i \in S$, то $D(i) = \min\limits_{\substack{j \in M \\ \{j,i\} \in N}} (D(j) + l(\{j,i\}))$.

Вначале $S' = \{1\}$ при $D'(1) = 0$ и $D(i) = +\infty$ для всех вершин, отличных от 1, т.е. $S = M \setminus \{1\}$.

(У105) Пусть $j \in S$, причем $D(j)$ является наименьшим среди всех $D(i)$, $i \in S$. Тогда $D(j)$ — минимальное расстояние от 1 до j. Таким образом, если $D(j) = \min\limits_{i \in S} D(i)$, то $D(j) = D'(j)$. [[10, стр. 188]]

Алгоритм:

Шаг 1. $D(1) = 0$, $S = \{2, \ldots, n\}$;

Шаг 2. для всех $i \in S$: если $\{1,\ i\} \in N$, то $D(i) = l(\{1,\ i\})$, иначе $D(i) = +\infty$;

Шаг 3. выбрать $j \in S$ такое, что $D(j) = \min\limits_{i \in S} D(i)$, $S = S \setminus \{j\}$;

Шаг 4. если $j = k$, то кратчайшее расстояние равно $D(j)$;

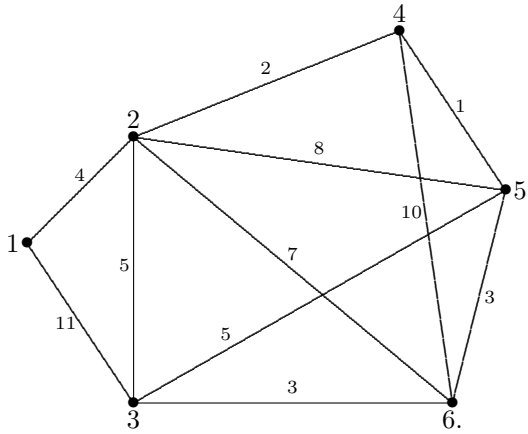
Шаг 5. для всех $i \in S$: если $\{j,i\} \in N$, то $D(i) = \min(D(i), D(j) + l(\{j,\ i\}))$. Переход к шагу 3.

(П) Рассмотрим граф $\Gamma = (M,\ N)$, $M = \{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$, N и l определяются таблицей,

$\{i,j\} \in N$	$l(\{i,j\})$
{1,2}	4
{1,3}	11
{2,3}	5
{2,4}	2
{2,5}	8
{2,6}	7
{3,5}	5
{3,6}	3
{4,5}	1
{4,6}	10
{5,6}	3

,

или геометрической реализацией,



Нужно найти кратчайший путь между вершинами 1 и 6.

Шаг 1. $D(1) = 0$, $S = \{2, 3, 4, 5, 6\}$;

Шаг 2. $D(2) = l(\{1, 2\}) = 4$, $D(3) = l(\{1, 3\}) = 11$, $D(4) = D(5) = D(6) = +\infty$;

Шаг 3. $\min\limits_{i \in S} D(i) = D(2)$, $j = 2$, $S = S \setminus \{2\} = \{3, 4, 5, 6\}$;

Шаг 4. $j \neq 6$;

Шаг 5. $D(3) = \min(D(3), D(2) + l(\{2, 3\})) = \min(11, 4 + 5) = 9$,
$D(4) = \min(D(4), D(2) + l(\{2, 4\})) = \min(+\infty, 4 + 2) = 6$,
$D(5) = \min(D(5), D(2) + l(\{2, 5\})) = \min(+\infty, 4 + 8) = 12$,
$D(6) = \min(D(6), D(2) + l(\{2, 6\})) = \min(+\infty, 4 + 7) = 11$;

Шаг 3. $\min\limits_{i \in S} D(i) = D(4)$, $j = 4$, $S = S \setminus \{4\} = \{3, 5, 6\}$;

Шаг 4. $j \neq 6$;

Шаг 5. $D(5) = \min(D(5), D(4) + l(\{4, 5\})) = \min(12, 6 + 1) = 7$,
$D(6) = \min(D(6), D(4) + l(\{4, 6\})) = \min(11, 6 + 10) = 11$;

Шаг 3. $\min\limits_{i \in S} D(i) = D(5)$, $j = 5$, $S = S \setminus \{5\} = \{3, 6\}$;

Шаг 4. $j \neq 6$;

Шаг 5. $D(3) = \min(D(3), D(5) + l(\{5, 3\})) = \min(9, 7 + 5) = 9$,
$D(6) = \min(D(6), D(5) + l(\{5, 6\})) = \min(11, 7 + 3) = 10$;

Шаг 3. $\min\limits_{i \in S} D(i) = D(3)$, $j = 3$, $S = S \setminus \{3\} = \{6\}$;

Шаг 4. $j \neq 6$;

Шаг 5. $D(6) = \min(D(6), D(3) + l(\{3, 6\})) = \min(10, 9 + 3) = 10$;

Шаг 3. $\min\limits_{i \in S} D(i) = D(6)$, $j = 6$, $S = S \setminus \{6\} = \varnothing$;

Шаг 4. $j = 6$, минимальное расстояние между вершинами 1 и 6 равно 10.

Предмет математической логики

Впервые формализовать методы мышления пытался величайший

древнегреческий философ Аристотель (IV в. до н.э.). Древнегреческий математик Евклид (III в. до н.э.) в своем знаменитом труде "Начала" при описания геометрии использовал аксиоматический метод: сначала приводились несколько аксиом (очевидных соотношений) и постулатов (недоказуемых простейших свойств), а все остальные свойства (теоремы) доказывались при помощи аксиом и уже доказанных теорем. Геометрии Евклида получилась настолько удачной, что дошла практически без изменений до современных курсов по геометрии (в школах). В XVII веке великий немецкий ученый, один создателей дифференциального исчисления, Лейбниц попытался описать всю математику, используя аксиоматический метод. Но его постигла неудача. Как было доказано только в XX веке, аксиоматический метод не подходит даже для описания всех свойств натуральных чисел.

Современная математическая логика, как самостоятельный раздел математики, сформировалась сравнительно недавно — на рубеже XIX–XX веков. Возникновение и быстрое развитие математической логики в начале XX века связано с так называемым кризисом в основаниях математики. Открытие парадоксов в теории множеств (например, парадокса Рассела-Цермело) привлекло к вопросам оснований математики практически всех ведущих математиков того времени. Известнейший немецкий математик Давид Гильберт предложил проводить формализацию математики, придерживаясь нескольких принципов. Важнейшим из этих принципов является принцип доказательной внутренней непротиворечивости такой формализации, т.е. наличие в системе формализации средств для доказательства этой непротиворечивости. Основные разделы математической логики — это исчисление высказываний и исчисление предикатов. А такой раздел математики как теория алгоритмов также обычно рассматривается как часть математической логики.

Логика высказываний

Язык L_0 логики высказываний состоит из:

1) Специальных символов: $\neg$, $\vee$, $\&$, $\Rightarrow$, (и);
2) Счетного множества переменных, обозначающих элементарные высказывания (атомарные формулы);
3) Если A и B — формулы, то $(A \vee B)$, $(A \,\&\, B)$, $(A \Rightarrow B)$ и $\neg A$ тоже формулы.

Далее запись формул будет упрощаться по правилам, подобным принятым в алгебре логики: использовать приоритет операций ($\neg$, $\&$, $\vee$, $\Rightarrow$), законы ассоциативности и опускать ненужные скобки.

Каждое элементарное высказывание обладает **семантикой**: значением истина или ложь.

Интерпретация I формулы — это сопоставление формуле зна-

чения истина (*И*) или ложь (*Л*) следующей последовательностью правил:

1) Каждой переменной сопоставляется значение либо *Л*, либо *И*;
2) Значение полученной формулы без переменных вычисляется как значение соответствующей ей булевой функции (0 соответствует *Л*, а 1 — *И*).

Формула A называется **истинной** в интерпретации I, если $I(A) = \mathit{И}$.

Формула A называется **общезначимой** или **тавтологией**, если $\forall I\ I(A) = \mathit{И}$. Обозначение: $\models A$.

Две формулы A и B **эквивалентны**, если $\models (A \Rightarrow B)\ \&\ (B \Rightarrow A)$. Эквивалентность формул будем обозначать $A \sim B$. Приоритет операции $\sim$ установим самым низким.

(П) Для любых формул A и B следующие формулы общезначимы:

1. $A\ \&\ \neg A \Rightarrow B$ — закон противоречия;
2. $\neg\neg A \Rightarrow A$ — закон снятия отрицания;
3. $A \Rightarrow \neg\neg A$ — закон навешивания отрицания;
4. $A \vee \neg A$ — закон исключения третьего.
5. $A \vee B \sim \neg A \Rightarrow B$ — удаление связки ИЛИ;
6. $A\ \&\ B \sim \neg(A \Rightarrow \neg B)$ — удаление связки И.

Формула Φ является **логическим следствием** множества формул Γ, если для любой интерпретации I, в которой истинны все формулы из Γ, истинна также и Φ: $\Gamma \models \Phi \quad \Leftrightarrow \quad \forall I((\forall B \in \Gamma\ I(B) = \mathit{И}) \Rightarrow (I(\Phi) = \mathit{И}))$. Если $\Gamma = \{\Phi_1, \ldots, \Phi_n\}$, то $\Gamma \models \Phi \quad \Leftrightarrow \quad \Phi_1, \ldots, \Phi_n \models \Phi \quad \Leftrightarrow \quad \models \Phi_1\ \&\ \cdots\ \&\ \Phi_n \Rightarrow \Phi$.

(П) Известно, что из Мюллера, Штирлица и Айсмана — двое или все трое немцы. Мюллер никогда не врет. Он говорит Штирлицу: “Вы такой же русский, как я немец”. Запишем эти высказывания в виде формул, используя элементарные высказывания МН, АН и ШН: Φ_1=(МН&АН)$\vee$(МН&ШН)$\vee$(АН&ШН); Φ_2=(МН$\Rightarrow\neg$ШН)&($\neg$ШН$\Rightarrow$МН). Составим таблицу истинности для этих формул:

АН	МН	ШН	Φ_1	Φ_2	
Л	*Л*	*Л*	*Л*	*Л*	
Л	*Л*	*И*	*Л*	*И*	
Л	*И*	*Л*	*Л*	*И*	
Л	*И*	*И*	*И*	*Л*	
И	*Л*	*Л*	*Л*	*Л*	
И	*Л*	*И*	*И*	*И*	+
И	*И*	*Л*	*И*	*И*	+
И	*И*	*И*	*И*	*Л*	

.

Получаем, что из истинности Φ_1 и Φ_2 следует истинность АН: $\Gamma = \{\Phi_1, \Phi_2\}$, $\Gamma \models$ АН.

Исчисление высказываний

Классическим исчислением высказываний называется набор из трех компонент:

1) язык L_0;
2) множество аксиом:
   A1) $A \Rightarrow (B \Rightarrow A)$;
   A2) $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$;
   A3) $(A \Rightarrow B) \Rightarrow ((A \Rightarrow \neg B) \Rightarrow \neg A)$;
   A4) $\neg\neg A \Rightarrow A$;
   A5) $A \Rightarrow (B \Rightarrow A \,\&\, B)$;
   A6) $A \,\&\, B \Rightarrow A$;
   A7) $A \,\&\, B \Rightarrow B$;
   A8) $A \Rightarrow A \vee B$;
   A9) $B \Rightarrow A \vee B$;
   A10) $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$ — эти выражения не формулы, а схемы бесконечного множества формул, получаемых из этих схем подстановкой в них вместо символов A, B и C любых формул;
3) правила вывода modus ponens* (m.p.) $\dfrac{A \Rightarrow B \quad A}{B}$ — про формулу B будем говорить, что она получается по правилу вывода m.p. из формул A и $A \Rightarrow B$.

Могут быть и другие исчисления высказываний, с другими наборами аксиом и функций. Необходимо только, чтобы булевы функции исчисления высказываний являлись полной системой и аксиомы при любой подстановке превращались в общезначимые формулы.

Пусть Γ — множество гипотез (постулатов). **Выводом** формулы A из Γ называется последовательность формул $A_1, \ldots, A_n$, где $A_n = A$ и $\forall i$ A_i — либо гипотеза, либо аксиома, либо выводима из A_j и A_k (j, $k < i$) по правилу m.p. Обозначение: $\Gamma \vdash A$, знак $\vdash$ называется штопор. Если $\Gamma = \{B_1, \ldots, B_m\}$, то можно писать $B_1, \ldots, B_m \vdash A$.

Если $\Gamma = \varnothing$ и $\Gamma \vdash A$, то A — **выводимая формула** или **теорема**. Обозначение — $\vdash A$.

(П) Формула $\Phi = A \Rightarrow A$ является теоремой в классическом исчислении высказываний, т.е. $\vdash A \Rightarrow A$, т.к. можно построить вывод:

1) подставим в аксиому A2 ($A \Rightarrow A$) вместо B и A вместо C, получим $\Phi_1 = \text{A2}\{(A \Rightarrow A)/B, A/C\} = (A \Rightarrow ((A \Rightarrow A) \Rightarrow A)) \Rightarrow ((A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A))$;

---

* правило отделения

2) подставим в аксиому A1 $(A \Rightarrow A)$ вместо B, получим $\Phi_2 = \text{A1}\{(A \Rightarrow A)/B\} = A \Rightarrow ((A \Rightarrow A) \Rightarrow A)$;
3) применим правило m.p. к Φ_1 и Φ_2 получим $\Phi_3 = (A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A)$;
4) подставим в аксиому A1 A вместо B, получим $\Phi_4 = \text{A1}\{A/B\} = A \Rightarrow (A \Rightarrow A)$;
5) применим правило m.p. к Φ_3 и Φ_4 получим $\Phi_5 = (A \Rightarrow A)$.

Φ_1, Φ_2, Φ_3, Φ_4, Φ_5 — это вывод формулы Φ из $\varnothing$.

(У106) Свойства штопора:

1) Из $\Gamma_1 \vdash A$ и $\Gamma_1 \subset \Gamma_2$ следует $\Gamma_2 \vdash A$ (в частности, из $\Gamma \vdash A$ следует $\Gamma, B \vdash A$);
2) Из $\Gamma_1 \vdash A$ и $\forall B \in \Gamma_1 (\Gamma_2 \vdash B)$ следует $\Gamma_2 \vdash A$;
3) $\Gamma \vdash A \quad \Leftrightarrow \quad \exists \Gamma_0 \subset \Gamma$, Γ_0 — конечно, $\Gamma_0 \vdash A$;
4) Из $\vdash A \Rightarrow B$ следует $A \vdash B$. [Очевидно]

Исчисление высказываний, для которого верно, что из $\vdash A$ следует $\vDash A$, называется **корректным**, т. е. любая теорема в корректном исчислении является общезначимой формулой.

Исчисление высказываний называется **непротиворечивым**, если для любой формулы A невозможно $\vdash A$ и $\vdash \neg A$.

Исчисление высказываний называется **полным**, если из $\vDash A$ следует $\vdash A$.

(У107) Классическое исчисление высказываний корректно, непротиворечиво и полно. [Корректность следует из того, что аксиомы — общезначимы, а правило m.p. сохраняет общезначимость. Непротиворечивость следует из корректности. Доказательство полноты см. в [13, стр. 69–70]]

(У108) **Теорема о дедукции.** В классическом исчислении высказываний из $\Gamma, A \vdash B$ следует $\Gamma \vdash A \Rightarrow B$. [[13, стр. 66]]

(П) Доказать $\vdash (A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C))$. Пусть 1) $A \Rightarrow B$, 2) $B \Rightarrow C$ и 3) A — гипотезы. Тогда применим m.p. к 1 и 3 получим 4) B. Затем применим m.p. к 2 и 4 получим 5) C. Следовательно, $A \Rightarrow B$, $B \Rightarrow C, A \vdash C$. Применив к последнему выражению теорему о дедукции 3 раза, докажем исходное.

(П) Докажем правило силлогизма, $A \Rightarrow B, B \Rightarrow C \vdash A \Rightarrow C$. Пусть $\Gamma = \{A \Rightarrow B, B \Rightarrow C\}$, тогда $\Gamma, A \vdash C$ (из A и $A \Rightarrow B$ выводится B, а из B и $B \Rightarrow C$ — C), что позволяет применить теорему о дедукции.

Исчисление предикатов 1-го порядка

Исчисление высказываний может использоваться только в очень ограниченном числе случаев. Рассмотрим, например, определение непрерывности функции f в точке x: “Для $\forall \varepsilon > 0$ $\exists \delta > 0$ такое, что $\forall y |x - y| < \delta$ выполняется $|f(x) - f(y)| < \varepsilon$”. Это же определение можно

записать так: $\forall\varepsilon(\varepsilon > 0 \Rightarrow \exists\delta(\delta > 0 \,\&\, \forall y(|x-y| < \delta \Rightarrow |f(x)-f(y)| < \varepsilon)))$. Записать это определение на языке логики высказываний не удаться, т.к. там, например, отсутствуют символы кванторов.

Алфавит языка L_1 логики предикатов 1-го порядка состоит из:

1) Специальных символов: $\forall\ \exists \Rightarrow \neg\ \&\ \vee\ (\ ,\ )$.
2) Счетного множества символов переменных;
3) Не более чем счетного множества символов констант;
4) Не более чем счетного множества функциональных символов;
5) Непустое и не более чем счетное множество предикатных символов.

Предикат 1-го порядка — это функция, определенная на множестве X^n (X — любое множество, а n — местность предиката) с область значений из двух элементов *И* (истина) и *Л* (ложь).

Предикат P, n-местный, однозначно определяется заданием подмножества Y множества X^n, $Y \subset X^n$:

$$P(a_1,\dots,a_n) = \begin{cases} \textit{И}, & \langle a_1,\dots,a_n\rangle \in Y; \\ \textit{Л}, & \langle a_1,\dots,a_n\rangle \notin Y. \end{cases}$$

Следовательно, бинарное отношение однозначно определяет двуместный предикат, а n-арное — n-местный.

Одноместный предикат называется **свойством**.

Нуль-местный предикат — это **высказывание**.

(П) Пусть предикат (свойство) $P(x)$ означает "x — простое число". Он истинен тогда и только тогда, когда x — простое число. Ясно, что, например, $P(1) = \textit{Л}$, $P(2) = \textit{И}$, $P(10) = \textit{Л}$, ...

Терм языка L_1 — это:

1) символ переменной или константы;
2) если f — n-местный функциональный символ и $t_1,\dots,t_n$ — термы, то $f(t_1,\dots,t_n)$ — терм.

Если P n-местный предикатный символ из алфавита L_1 и $t_1,\dots,t_n$ — термы, то $P(t_1,\dots,t_n)$ — **атомарная формула** языка L_1. Если A и B — формулы языка L_1 и x символ переменной из алфавита L_1, то $(A \Rightarrow B)$, $(A\ \&\ B)$, $(A \vee B)$, $\neg A$, $\forall xA$ и $\exists xA$ — тоже формулы языка L_1. Далее запись формул будет упрощаться по правилам, подобным принятым в алгебре логики: использовать приоритет операций ($\forall, \exists, \neg, \&, \vee, \Rightarrow$), законы ассоциативности и опускать ненужные скобки.

В формуле $\forall xA$ или $\exists xA$ формула A называется **областью действия квантора**.

Вхождение переменной в формулу называется **связанным**, если оно находится в области действия квантора по этой переменной. В противном случае оно называется **свободным**. Использование квантора называется операцией **связывания квантором**.

Формула, не содержащая свободных переменных, называется **предложением**. Каждое предложение может иметь только одно из двух значений: истина (*И*) или ложь (*Л*).

Формулу A, содержащую свободные переменные $x_1, \ldots, x_n$, будем обозначать $A(x_1, \ldots, x_n)$.

Областью интерпретации называется непустое множество D, которое состоит из сущностей, соответствующих символам констант и переменных языка L_1.

(П) Пусть $D = \mathbb{N}$ и пусть предикат $P(x)$ означает "x — четное", $Q(x)$ — "x делится на 3". Тогда выражение $P(x)$ & $Q(x)$ означает "x делится на 6". Предложение $\exists x(P(x)\ \&\ Q(x))$ — истинно, $\forall x(P(x)\ \&\ Q(x))$ — ложно, а формула $\exists x(P(x)\ \&\ Q(y))$ не есть предложение, т.к. переменная y свободна. В качестве возможных значений для x здесь подразумевались натуральные числа.

Интерпретацией I формулы A, обозначение — $I(A)$, называется следующая последовательность сопоставлений:

1) каждому символу константы сопоставляется конкретное значение из D;
2) каждому n-местному функциональному символу f сопоставляется функция $f\colon D^n \to D$;
3) каждому n-местному предикатному символу P сопоставляется предикат $P\colon D^n \to \{И, Л\}$;
4) за каждым символом переменной закрепляется множество D ее возможных значений.

Значение терма t на значениях $b_1, \ldots, b_m \in D$ переменных $x_1, \ldots, x_m$ обозначается $t[b_1, \ldots, b_m]$.

Значение формулы $A(x_1, \ldots, x_m)$ в интерпретации I на наборе $b_1, \ldots, b_m \in D$ значений своих свободных переменных обозначается $I(A)[b_1, \ldots, b_m]$.

(П) Рассмотрим формулу $A = \forall x_1 \exists x_2 P(x_1, x_2) \Rightarrow \exists x_2 \forall x_1 P(x_1, x_2)$ и интерпретацию I, в которой сопоставим P отношение "меньше или равно", а D — множество $\{0, 1\}$. Составим таблицу значений предиката $\leqslant$:

x_1	x_2	$\leqslant$
0	0	*И*
0	1	*И*
1	0	*Л*
1	1	*И*

.

Следовательно, $I(\forall x_1 \exists x_2 P(x_1,\ x_2)) = И$, $I(\exists x_2 \forall x_1 P(x_1,\ x_2)) = И$ и $I(A) = И$ (ведь в D существует $x_2 = 1$).

(П) Рассмотрим предыдущую формулу в измененной интерпретации I, в которой сопоставим P отношение "равно":

x_1	x_2	$=$
0	0	*И*
0	1	*Л*
1	0	*Л*
1	1	*И*

Здесь посылка верна, но следствие ложно. Следовательно, $I(A) = Л$.

Формула $A(x_1, \ldots, x_n)$ называется **выполнимой**, если в некоторой интерпретации I существуют $b_1, \ldots, b_n \in D$ такие, что

$$I(A)[b_1, \ldots, b_n] = И.$$

Формула $A(x_1, \ldots, x_n)$ называется **истинной** в интерпретации I, если для любых $b_1, \ldots, b_n \in D$ $I(A)[b_1, \ldots, b_n] = И$. Обозначение $I(A) = И$.

Формула $A(x_1, \ldots, x_n)$ называется **общезначимой**, если в любой интерпретации I она истинна. Обозначение $\models A$.

(У109) Формула A общезначима тогда и только тогда, когда формула $\neg A$ невыполнима, и формула A выполнима тогда и только тогда, когда $\neg A$ необщезначима. [Очевидно]

(П) Формула $A(x) = \exists y P(x, y)$ выполнима и истинна в интерпретации I, которой соответствует $D = \{0, 1\}$ и предикат "меньше или равно", но она не общезначима. Эта же формула в интерпретации с предикатом "$<$" выполнима ($I(A)[0] = И$), но не истинна.

(У110) $\forall x A \sim \neg\exists x \neg A$ (связку $\sim$ используем так же, как и в логике высказываний). [Нужно доказать $\models \forall x A \Rightarrow \neg\exists x \neg A$ и $\models \neg\exists x \neg A \Rightarrow \forall x A$. Докажем 1-е соотношение. Предположим противное. Пусть в некоторой интерпретации I посылка истинна, а заключение ложно, т. е. $I(\forall x A) = И$ и $I(\neg\exists x \neg A) = Л$. Тогда $I(\exists x \neg A) = И$, т. е. существует такое значение b_0 x, что $I(A)[b_0] = Л$, но это противоречит истинности посылки. Таким же образом доказывается и 2-е соотношение. Если $I(\neg\exists x \neg A) = И$ и $I(\forall x A) = Л$, то $I(\exists x \neg A) = Л$, т. е. не существует такое значение b_0 x, что $I(\neg A)[b_0] = И$ или $I(A)[b_0] = Л$, но это противоречит ложности заключения. Соотношение $\exists x A \sim \neg\forall x \neg A$ доказывается аналогично.]

Терм t называется **свободным для переменной** x в $A(x)$, если никакое свободное вхождение x в $A(x)$ не находится в области действия квантора по любой переменной, входящей в t.

(П) В формуле $A(x) = \exists y(x < y)$ терм y или $y^2 * w$ не свободен для x, но свободен, например, для термов z или $w^2 * v$.

Интерпретация, в которое истинны все формулы из множества Γ, называется **моделью** Γ, J_Γ.

Формула A — **логическое следствие** из множества формул Γ, если при любой интерпретации множества $\Gamma \cup \{A\}$, формула A выполнима на любых наборах значений из D, на которых выполнимы все формулы из Γ. Обозначение: $\Gamma \vDash A$. Если $\Gamma = \{A_1, \ldots, A_k\}$, то можно также писать $A_1, \ldots, A_k \vDash A$ или $\vDash A_1 \,\&\, \cdots \,\&\, A_k \Rightarrow A$.

(П) Рассмотрим формулы $\Phi_1 = \forall x P(x, x)$ и $\Phi_2 = \exists y \forall x P(x, y)$. Если $\Gamma = \{\Phi_1, \Phi_2\}$ и $A = \forall x \exists y P(x, y)$, то $\Gamma \vDash A$, т.к. в A достаточно взять $y = x$.

Понятие о предикатах высокого порядка

Предикат n-го порядка допускает в качестве своих аргументов предикаты порядка, меньшего n. Предикаты порядка выше 1-го называются **предикатами высокого порядка**.

(П) Рассмотрим предикат ДЕЛИМОСТЬ(x, y), который означает, что x делится на y. Например, ДЕЛИМОСТЬ(6, 3) = *И*, ДЕЛИМОСТЬ(12, 4) = *И*, ДЕЛИМОСТЬ(19, 7) = *Л*, предложение $\forall x \exists y$ДЕЛИМОСТЬ(x, y) — истинно. Теперь предположим, что требуется узнать, существует ли какое-нибудь отношение, связывающее числа 51 и 17, т.е. проверить истинно ли предложение $\exists z(z(51, 17))$, в котором квантор используется по переменной-предикату. Это предложение можно записать и как $\exists z$(ОТНОШЕНИЕ$(z, 51, 17)$). В интерпретации, в области которой есть предикат ДЕЛИМОСТЬ, оно истинно. Предикат ОТНОШЕНИЕ имеет 2-й порядок.

Исчисление предикатов 1-го порядка

Классическим исчислением предикатов 1-го порядка называется набор из трех компонент:

1) язык L_1;
2) множество аксиом:

A1–A10) из классического исчисления высказываний;

A11) $\forall x A(x) \Rightarrow A(t)$, здесь и в A12 t свободен для x в $A(x)$;

A12) $A(t) \Rightarrow \exists x A(x)$;

3) правил вывода:

R1) modus ponens (m.p.);

R2) связывание квантором общности: $\dfrac{B \Rightarrow A(x)}{B \Rightarrow \forall x A(x)}$, здесь и в R3 x не входит свободно в B;

R3) связывание квантором существования: $\dfrac{A(x) \Rightarrow B}{\exists x A(x) \Rightarrow B}$.

Выводом формулы A из множества гипотез (постулатов) называется последовательность формул $B_1, \ldots, B_n$, в которой $B_n = A$, а B_i $(i \leqslant n)$ — это либо аксиома, либо гипотеза, либо формула, полученная из предыдущих формул по правилам вывода, причем ни одно из применений правил R2 и R3 к формуле, зависящей от гипотезы, не связывает переменную, свободно входящую в эту гипотезу. Обозначения остаются такими же как и в логике высказываний.

Формула B_i **зависит от гипотезы** $B \in \Gamma$ в выводе, если:

1) $B_i = B$, т.е. B_i обосновывается как гипотеза из Γ;
2) B_i получается по правилу вывода m.p. из B_k и B_j $(j, k < i)$ и B_k или B_j зависит от B.

(У111) Аксиомы исчисления предикатов — это общезначимые формулы. [[13, стр. 87]]

(У112) Правила вывода R1–R3 сохраняют общезначимость формул. [[13, стр. 87]]

(У113) Классическое исчисление предикатов 1-го порядка корректно и непротиворечиво. [Корректность следует из двух предыдущих утверждений, а непротиворечивость — из корректности.]

(У114) **Теорема Геделя о полноте.** Классическое исчисление предикатов 1-го порядка является полным, т. е. из $\vDash A$ следует $\vdash A$. [[6, стр. 224–225]]

(У115) **Теорема о дедукции.** Если $\Gamma, A \vdash B$, то $\Gamma \vdash A \Rightarrow B$. [[13, стр. 86]]

(П) Рассмотрим вывод из $\Gamma = \{P(x)\}$:

1) $\Gamma \vdash P(x)$;
2) $\vdash P(x) \Rightarrow ((A \Rightarrow A) \Rightarrow P(x))$ (A1);
3) $\Gamma \vdash (A \Rightarrow A) \Rightarrow P(x)$ (m.p. 1,2);
4) $\Gamma \vdash (A \Rightarrow A) \Rightarrow \forall x P(x)$ (R2);
5) $\vdash A \Rightarrow A$ (известно);
6) $P(x) \vdash \forall x P(x)$ (m.p. 4,5);
7) $\vdash P(x) \Rightarrow \forall x P(x)$.

Но формула $B = P(x) \Rightarrow \forall x P(x)$ необщезначима, т.к. в интерпретации с $D = \{0, 1\}$ и предикатом P таким, что $I(P(0)) = Л$ и $I(P(1)) = И$, $I(B)[1] = Л$. Этот пример показывает, что нельзя связывать свободные переменные в формуле, зависящей от гипотезы.

Формула A находится в **предваренной нормальной форме**, если A имеет вид $Q_1 x_1 \ldots Q_n x_n B$, где Q_i $(1 \leqslant i \leqslant n)$ — это квантор, а B не содержит кванторов. Формулу B называют **матрицей** A, а запись $Q_1 x_1 \ldots Q_n x_n$ — **префиксом** A.

(У116) Верны следующие соотношения:

1. $\neg \exists x A(x) \sim \forall x \neg A(x)$;

2. $\neg\forall x A(x) \sim \exists x \neg A(x)$;
3. $A \,\&\, \exists x B(x) \sim \exists x(A \,\&\, B(x))$ — здесь и далее x не входит свободно в A;
4. $A \vee \forall x B(x) \sim \forall x(A \vee B(x))$;
5. $A \,\&\, \forall x B(x) \sim \forall x(A \,\&\, B(x))$;
6. $A \vee \exists x B(x) \sim \exists x(A \vee B(x))$;
7. $A \Rightarrow \exists x B(x) \sim \exists x(A \Rightarrow B(x))$;
8. $A \Rightarrow \forall x B(x) \sim \forall x(A \Rightarrow B(x))$;
9. $\forall x B(x) \Rightarrow A \sim \exists x(B(x) \Rightarrow A)$;
10. $\exists x B(x) \Rightarrow A \sim \forall x(B(x) \Rightarrow A)$. [Очевидно]

(У117) Для любой формулы логики предикатов можно построить эквивалентную ей формулу, находящуюся в предваренной нормальной форме. [Конструктивное — алгоритмом:

Шаг 1. Переименовать связанные переменные так, чтобы ни одна переменная не была одновременной связанной и свободной;

Шаг 2. Переименовать связанные переменные так, чтобы переменные, связанные разными кванторами, имели разные имена;

Шаг 3. Применять правила замен 1–10 до тех пор, пока кванторы не перейдут в префикс.]

(П) Привести к предваренной нормальной форме формулу

$$(\forall x A(x) \Rightarrow \forall x B(x)) \Rightarrow \exists x C(x, y) \,\&\, D(x) :$$

1) $(\forall x_1 A(x_1) \Rightarrow \forall x_1 B(x_1)) \Rightarrow \exists x_1 C(x_1, y) \,\&\, D(x)$;
2) $(\forall x_1 A(x_1) \Rightarrow \forall x_2 B(x_2)) \Rightarrow \exists x_3 C(x_3, y) \,\&\, D(x)$;
3) $\exists x_3 \forall x_1 \exists x_2((A(x_1) \Rightarrow B(x_2)) \Rightarrow C(x_3, y) \,\&\, D(x))$.

Формула в ДНФ имеет вид $C_1 \vee \cdots \vee C_n$, где C_i — это конъюнкция атомарных формул или их отрицаний.

Формула в КНФ имеет вид $C_1 \,\&\, \cdots \,\&\, C_n$, где C_i — это дизъюнкция атомарных формул или их отрицаний, $n > 0$. C_i называют **дизъюнктами**.

(У118) По каждой формуле, не содержащей кванторов, можно построить эквивалентную ей формулу, находящуюся в ДНФ или КНФ. [Используем по-порядку следующие соотношения:

1. $C \Rightarrow D \sim \neg C \vee A$ — убираем импликацию;
2. $\neg(C \vee D) \sim \neg C \,\&\, \neg D$, $\neg(C \,\&\, D) \sim \neg C \vee \neg D$ — сдвигаем $\neg$ к атомарным формулам;
3. $\neg\neg C \sim C$ — уничтожаем лишние $\neg$;
4. $C \vee (D \,\&\, E) \sim (C \vee D) \,\&\, (C \vee E)$ — для КНФ;
5. $C \,\&\, (D \vee E) \sim (C \,\&\, D) \vee (C \,\&\, E)$ — для ДНФ;
6. $C \,\&\, D \sim D \,\&\, C$, $C \vee D \sim D \vee C$ — используются вместе с 5 и 4.

Описанный алгоритм строит формулу в КНФ или ДНФ, эквивалентную исходной.]

(П) Привести к КНФ и ДНФ формулу $(A \Rightarrow B) \vee \neg(C \,\&\, A)$: $(\neg A \vee B) \vee \neg(C \,\&\, A) \sim \neg A \vee B \vee \neg C \vee \neg A$ — это и КНФ и ДНФ.

(П) Привести к КНФ и ДНФ формулу $(B \,\&\, C \Rightarrow A) \,\&\, (A \vee C \Rightarrow \neg B) \vee \neg(\neg A \,\&\, B)$: $(\neg(B \,\&\, C) \vee A) \,\&\, (\neg(A \vee C) \vee \neg B) \vee \neg(\neg A \,\&\, B) \sim (\neg B \vee \neg C \vee A) \,\&\, (\neg A \,\&\, \neg C \vee \neg B) \vee \neg\neg A \vee \neg B \sim (\neg B \vee \neg C \vee A) \,\&\, (\neg A \,\&\, \neg C \vee \neg B) \vee A \vee \neg B \sim$

(КНФ) $\sim (A \vee \neg B) \vee (\neg B \vee \neg C \vee A) \,\&\, (\neg A \,\&\, \neg C \vee \neg B) \sim (A \vee \neg B) \vee (\neg B \vee \neg C \vee A) \,\&\, (\neg B \vee \neg A \,\&\, \neg C) \sim (A \vee \neg B \vee \neg B \vee \neg C \vee A) \,\&\, ((A \vee \neg B \vee \neg B) \vee \neg A \,\&\, \neg C) \sim (A \vee \neg B \vee \neg B \vee \neg C \vee A) \,\&\, (A \vee \neg B \vee \neg B \vee \neg A) \,\&\, (A \vee \neg B \vee \neg B \vee \neg C)$

(ДНФ) $\sim \neg A \,\&\, \neg C \,\&\, \neg B \vee \neg A \,\&\, \neg C \,\&\, \neg C \vee \neg A \,\&\, \neg C \,\&\, A \vee \neg B \,\&\, \neg B \vee \neg B \,\&\, \neg C \vee \neg B \,\&\, A \vee A \vee \neg B$.

Метод резолюций

Важнейшая проблема логики предикатов — это проверка формул на общезначимость. Другая важная проблема состоит в следующем. Дано множество формул Γ и формула Φ и нужно проверить, является ли Φ логическим следствием Γ. Последняя проблема сводится к первой, т.к. если $\Gamma = \{\Phi_1, \ldots, \Phi_n\}$, то $\Gamma \models \Phi \quad \Leftrightarrow \quad \models \Phi_1 \,\&\, \cdots \,\&\, \Phi_n \Rightarrow \Phi$.

В 1965 году Дж. Робинсон опубликовал описание принципа вывода, позволяющего во многих случаях решать описанные проблемы. Этот принцип вывода получил название **метод резолюций**. Метод резолюций легко формализуется для использования на ЭВМ. Он стал основой языка программирования ПРОЛОГ. В проекте компьютеров 5-го поколения (Япония) его предполагается реализовать на аппаратном уровне.

Общезначимость формулы A методом резолюций доказывается путем доказательства невыполнимости формулы $\neg A$. В качестве формул можно рассматривать только предложения. Метод резолюций можно применять только к формулам специального вида. Приведение формул к такому виду осуществляется следующей последовательностью шагов:

Шаг 1. Для заданной формулы $A_0 = \neg A$ строится эквивалентная ей предваренная нормальная форма $A_1 = Q_1 x_1 \ldots Q_n x_n B$. Невыполнимость A_0 равнозначна невыполнимости A_1.

Шаг 2. Для матрицы B формулы A_1 строим эквивалентную ей КНФ $B_1 = C_1 \,\&\, \cdots \,\&\, C_k$, где C_i — это дизъюнкты. Получаем $A_2 = Q_1 x_1 \ldots Q_n x_n (C_1 \,\&\, \cdots \,\&\, C_k)$. Невыполнимость A_1 равнозначна невыполнимости A_2.

Шаг 3. Убираем из формулы A_2 все кванторы существования. Пусть $Q_r = \exists$ и $Q_s = \forall$ для всех $s < r$. Тогда можно убрать из префикса A_2 $\exists x_r$ и заменить в матрице A_2 все вхождения x_r

на $f_r(x_1, \ldots, x_{r-1})$. Повторяем аналогичную процедуру до тех пор, пока не уберем все кванторы $\exists$. В случае, когда префикс формулы начинается с квантора существования, заменяем все вхождения x_r на константу a_r. Получим формулу A_3.

Шаги 2 и 3 можно менять местами.

Предваренную нормальную форму некоторой формулы, избавленную описанным в шаге 3 образом от кванторов $\exists$, называют **сколемовской нормальной формой** этой формулы.

Сколемовская нормальная форма, матрица которой — КНФ, называется **клаузальной формой**.

Функции f_r и константы a_r называются соответственно **сколемовскими функциями** и **сколемовскими константами**.

В сколемовской нормальной или в клаузальной формах все переменные связаны квантором общности, поэтому в них можно условно опускать префикс.

(П) Формула $\forall x \exists y \forall z P(x, y, z)$ означает, что для любых x существует y такой, что $P(x, y, z)$ истинно для всех z. Отсюда следует, что y зависит только от x или $y = f(x)$. Следовательно сколемовской нормальной формой для нее будет $P(x, f(x), z)$.

(П) Для формулы $\forall x \forall y \exists z P(x, y, z)$ сколемовской нормальной формой будет $P(x, y, f(x, y))$.

(П) Сколемовской нормальной формой формулы

$$\exists v \forall w \forall x \exists y \exists z P(v, w, x, y, z)$$

является $P(a, w, x, f(w, x), g(w, x))$, где a — сколемовская константа.

(У119) Невыполнимость формулы A_0 равнозначна невыполнимости ее клаузальной формы A_3. [[18, стр. 75]]

Множество дизъюнктов клаузальной формы $A_3 = C_1 \,\&\, \cdots \,\&\, C_k$ формулы A_0 называется ее **клаузальным множеством**. Обозначение $S = \{C_1, \ldots, C_k\}$. Далее будем автоматически переносить терминологию, применимую к клаузальным формам, на клаузальные множества, т. е. будем говорить о выполнимости, истинности или ложности в заданной интерпретации и т.п. клаузальных множеств.

(П) Построим клаузальное множество для формулы

$$A_0 = \neg \exists x_1 \forall x_2 \exists x_3 \forall x_4 (P(x_1, x_2, x_3) \Rightarrow P(x_2, x_3, x_4)) :$$

$$\forall x_1 \exists x_2 \forall x_3 \exists x_4 \neg (P(x_1, x_2, x_3) \Rightarrow P(x_2, x_3, x_4)) \sim$$

$$\sim \forall x_1 \exists x_2 \forall x_3 \exists x_4 \neg (\neg P(x_1, x_2, x_3) \vee P(x_2, x_3, x_4)) \sim$$

$$\sim \forall x_1 \exists x_2 \forall x_3 \exists x_4 (P(x_1, x_2, x_3) \ \& \ \neg P(x_2, x_3, x_4))$$

$$\forall x_1 \forall x_3 (P(x_1, f(x_1), x_3) \ \& \ \neg P(f(x_1), x_3, g(x_1, x_3)));$$

$S = \{P(x_1, f(x_1), x_3), \neg P(f(x_1), x_3, g(x_1, x_3)\}$.

(У120) Невыполнимость хотя бы одного дизъюнкта из клаузального множества формулы $\neg A$ означает общезначимость формулы A. [Очевидно]

Эрбрановским универсумом, соответствующим клаузальному множеству S, называется множество H такое, что:

1. в H входят все константы из S, а если в S нет констант, то в H добавляется произвольная константа;
2. если $t_1, \ldots, t_k \in H$ и f — k-местный функциональный символ из S, то $f(t_1, \ldots, t_k) \in H$.

(П) Для S из предыдущего примера

$$H = \{a, f(a), f(f(a)), g(a, a), f(g(a, a)), g(a, f(a)), \ldots\},$$

где a — константа. Этот пример показывает, что H — это множество термов и что если в S входит хотя бы один функциональный символ, то H содержит бесконечное число элементов.

Атомарные формулы вида $P(t_1, \ldots, t_k)$, где P — это предикатный символ из S, а $t_1, \ldots, t_k \in H$ — называются **основными атомами**.

Эрбрановской или **H-интерпретацией** называется интерпретация, для которой область интерпретации $D = H$; каждому символу константы сопоставляется константа из H, каждому n-местному функциональному символу сопоставляется n-местная функция, осуществляющая отображение из H^n в H, и каждому n-местному предикатному символу сопоставляется n-местный предикат, определенный на H^n.

Основной пример дизъюнкта C_i из S — это дизъюнкт C_i, у которого все переменные заменены на элементы эрбрановского универсума (термы из H).

Множество всех основных атомов клаузального множества S называется **эрбрановским базисом** S, B_H.

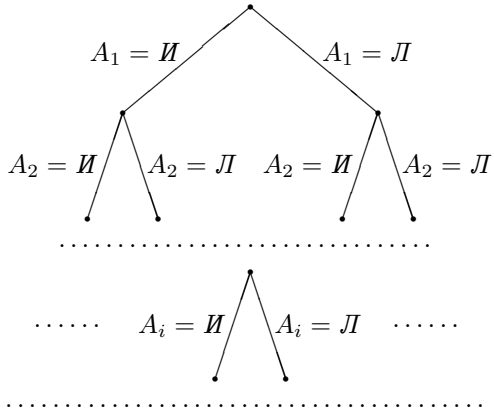
(П) Для S из предыдущего примера эрбрановским базисом будет множество $B_H = \{P(a,\ a,\ a),\ P(a,\ a,\ f(a)),\ P(a,\ a,\ f(f(a))),\ P(a,\ a,\ g(a,\ a)),\ P(a,\ f(a),\ f(g(a,\ a))),\ \ldots\}$, а основным примером дизъюнкта $P(x_1, f(x_1), x_3)$ будет $P(a, f(a), a)$ или $P(f(a), f(f(a)), g(a, a))$.

Если условиться обозначать константы символами a, b, c, d, e, функции — f, g, h, то можно использовать сокращения: $P(a,\ a,\ a) = Paaa$, $P(a, a, f(a)) = Paafa$, $P(a, a, g(a, a)) = Paag(aa)$, $P(f(a), f(f(a)), g(a, a)) = P(fa, ffa, gaa)$ и т.п. При использовании сокращений

необходимо избегать двусмысленностей, например, $Sfxx$ можно понять как $S(f(x), x)$ $[Sf(x)x]$, а можно и как $S(f(x, x))$.

(У121) Клаузальное множество S невыполнимо тогда и только тогда, когда S ложно во всех H-интерпретациях. [[18, стр. 78]]

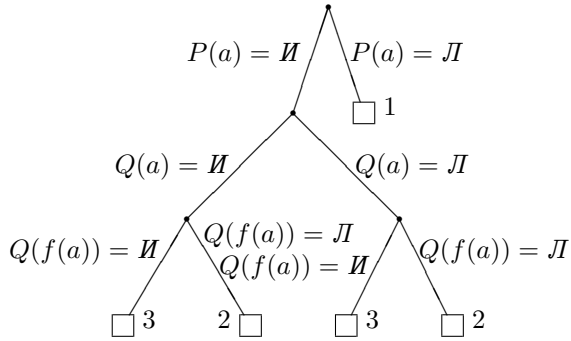
Семантическим деревом T для клаузального множества S с эрбрановским базисом $\{A_1, A_2, \ldots, A_i, \ldots\}$, называется бинарное дерево T вида:



— любой путь из его корня задает конкретную H-интерпретацию и любой H-интерпретации соответствует единственный путь из корня. Чтобы установить, что S невыполнимо, достаточно показать, что любой путь из вершины T соответствует H-интерпретации, в которой ложен хотя бы один основной пример дизъюнкта из S: не нужно знать весь этот (бесконечный) путь, достаточно знать ту его часть, которая определяет значение данного основного примера. Использование семантического дерева позволяет работать с предикатами как с высказываниями: в семантическом дереве каждому основному атому назначается значение либо истина, либо ложь, т.е. основные атомы интерпретируются здесь как высказывания.

Узел N семантического дерева T для клаузального множества S называется **опровергающим**, если в нем опровергается некоторый основной пример дизъюнкта из S и никакой из узлов, лежащих выше N, таким свойством не обладает.

(П) Для $S = \{P(x), \neg P(x) \vee Q(f(x)), \neg Q(f(a))\}$ эрбрановским базисом будет множество $\{P(a), Q(a), Q(f(a)), P(f(a)), \ldots\}$. Построим семантическое дерево для S:



— квадратами изображены все опровергающие узлы, двигаться дальше которых бессмысленно. Для доказательства невыполнимости S, нужно показать, что все пути из корня проходят через опровергающие узлы, что и сделано, т.е. доказано, что S — невыполнимо.

(У122) Пусть S — невыполнимое клаузальное множество. Тогда на любом пути из корня семантического дерева T, соответствующего S, лежит опровергающий узел и множество опровергающих узлов в T конечно. [1-е очевидно.]

(У123) **Теорема Эрбрана.** Клаузальное множество S невыполнимо тогда и только тогда, когда существует конечное невыполнимое множество основных примеров дизъюнктов из S. [Действительно, если S — невыполнимо, то в соответствующем ему семантическом дереве каждый путь из корня завершается в опровергающем узле и число таких узлов конечно. А т.к. в каждом опровергающем узле опровергается, как минимум, один основной пример дизъюнкта из S, то отсюда следует существование конечного невыполнимого множества основных примеров дизъюнктов из S. Далее предположим, что существует конечное невыполнимое множество основных примеров дизъюнктов из S и S — выполнимо. Невыполнимость множества основных примеров означает, что любой путь из корня в семантическом дереве должен заканчиваться опровергающим узлом, т.е. не существует H-интерпретации, в которой бы не опровергалось S, — противоречие.]

Смысл теорема Эрбрана в том, что она позволяет свести вопрос об общезначимости произвольного предложения языка логики предикатов 1-го порядка, к вопросу о невыполнимости конечного числа, построенных по нему, бескванторных формул. В ней формулируются общие правила для создания алгоритма для порождения подобных бескванторных формул. Теорема Эрбрана, в частности, лежит в основе метода резолюций.

Метод резолюций для логики высказываний

Предикат или предикат вместе с отрицанием называется **литерой**.

Пусть $C_1 = C_1' \vee p$ и $C_2 = C_2' \vee \neg p$ — дизъюнкты. Тогда формула $C_1' \vee C_2'$ называется **резольвентой** C_1 и C_2, а p и $\neg p$ — **отрезаемыми литерами**.

Правило вывода резольвенты $\dfrac{C_1' \vee p \quad C_2' \vee \neg p}{C_1' \vee C_2'}$ называется **правилом резолюции**. Применение правила резолюции к дизъюнктам p и $\neg p$ приводит к выводу пустого дизъюнкта, $\square = Л$.

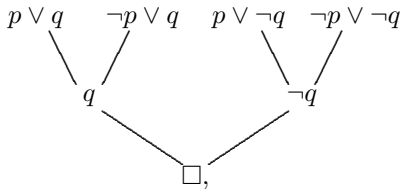
Правило вывода $\dfrac{C \vee p \vee p}{C \vee p}$ называется **правилом склейки**.

(У124) Резольвента дизъюнктов $C_1 = C_1' \vee p$ и $C_2 = C_2' \vee \neg p$ является их логическим следствием, т.е. $C_1, C_2 \vDash C_1' \vee C_2'$. []

Пусть S — клаузальное множество. **Резолютивным выводом** C из S называется последовательность $C_1, \ldots, C_n$, в которой $C_n = C$ и каждый C_i $(1 \leqslant i \leqslant n)$ либо входит в S, либо получен из предыдущих членов последовательности по правилу резолюции.

(У125) Из клаузального множества S резолютивно выводится $\square$ тогда и только тогда, когда S — невыполнимо. []

(П) $S = \{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$. Построим резолютивный вывод,



следовательно, S — невыполнимо.

Метод резолюций для логики предикатов 1-го порядка

Подстановка — это множество вида $\{t_1/x_1, \ldots, t_n/x_n\}$, где все x_i $(1 \leqslant i \leqslant n)$ — это различные символы переменных, а t_i — терм, не равный x_i. Будем обозначать подстановки маленькими греческими буквами θ, σ, λ.

Пусть C — дизъюнкт, а θ — подстановка. Тогда $C\theta$ — дизъюнкт, который получен из C одновременной заменой x_i на t_i. Подстановки можно применять и к отдельным термам.

(П) $S = \{C_1, C_2, C_3\}$, где $C_1 = P(x) \vee Q(x)$, $C_2 = \neg P(f(y))$, $C_3 = \neg Q(f(a))$. Нужно выяснить является ли S невыполнимым. S соответствует формула $\forall x \forall y (C_1 \,\&\, C_2 \,\&\, C_3)$. Если найти такие x и y, что $C_1 \,\&\, C_2 \,\&\, C_3 = Л$, то это докажет невыполнимость (ложность) S.

1) $\theta = \{f(a)/x, a/y\}$, $C_1\theta = P(f(a)) \vee Q(f(a))$, $C_2\theta = \neg P(f(a))$, т.е. θ

превращает дизъюнкты в их основные примеры (формулы логики высказываний).

2) $C_4 = Q(f(a))$ — по правилу резолюции к $C_1\theta$ и $C_2\theta$.

3) $C_5 = \square$ — по правилу резолюции к C_3 и C_4.

Доказано, что S — невыполнимо, т.к. при $x = f(a)$ и $y = a$ C_1 & C_2 & $C_3 = Л$.

Пусть $w = \{l_1, \ldots, l_n\}$ — множество литер. Тогда подстановка θ такая, что $l_1\theta = \cdots = l_n\theta$, называется **унификатором** w. Для унификатора θ верно, что $\#(w\theta) = 1$, где $w\theta = \{l_1\theta, \ldots, l_n\theta\}$.

Пусть $\theta = \{t_1/x_1, \ldots, t_n/x_n\}$ и $\lambda = \{s_1/y_1, \ldots, s_m/y_m\}$ — подстановки. **Композицией** подстановок θ и λ называется подстановка σ, получаемая из множества $\{t_1\lambda/x_1, \ldots, t_n\lambda/x_n, s_1/y_1, \ldots, s_m/y_m\}$ вычеркиванием всех элементов вида:

1. s_i/y_i, для которых справа от символа / в θ существует $x_k = y_i$;
2. $t_i\lambda/x_i$, если $t_i\lambda = x_i$.

Подстановка σ действует как последовательное применение подстановок θ и затем λ, т.е. $\sigma = \theta\lambda$.

(У126) Свойства композиции подстановок:

1. $C(\theta\lambda) = (C\theta)\lambda$;
2. $(\sigma\theta)\lambda = \sigma(\theta\lambda)$. []

Унификатор σ для множества литер w называется **наиболее общим** (НОУ), если для любого унификатора w λ верно, что существует подстановка θ такая, что $\lambda = \sigma\theta$.

(У127) Следующий алгоритм обеспечивает за конечное число шагов либо вычисление НОУ σ заданного множества литер $w = \{l_1, \ldots, l_s\}$, либо сообщает об отсутствии унификатора для w:

Шаг 1. Просмотр всех литер $l_1, \ldots, l_s$ — они должны иметь идентичную структуру, т.е. все предикатные символы должны быть одинаковы и все литеры должны быть либо с отрицанием, либо без отрицания. Если структура литер неидентична, то НОУ для w не существует.

Шаг 2. Пусть $n = 0$, $w_0 = w$, $\sigma_0 = \varnothing$ (пустая, ничего не меняющая подстановка).

Шаг 3. Если $\#w_n = 1$, то НОУ найден, $\sigma = \sigma_n$.

Шаг 4. Выделение первых слева позиций в литерах из w_n, где они различаются, и занесение из каждой литеры по терму, начинающемуся с этой позиции, в множество Δ_n, которое называется **множеством рассогласования**.

Шаг 5. Если в Δ_n есть символ переменной x и терм t, не содержащий x, то $\sigma_{n+1} = \sigma_n\{t/x\}$, $w_{n+1} = w_n\{t/x\}$, $n = n + 1$ и переход к шагу 3, иначе НОУ не существует.

[Алгоритм всегда заканчивает работу за конечное число шагов, т. к. w_{n+1} содержит по крайней мере на одну переменную меньше, чем w_n, а число переменных в литерах из w конечно. То, что в результате работы алгоритма получается именно НОУ принимаем без доказательства.]

(П) Пусть $w = \{P(x,\, f(x),\, y),\, P(f(z),\, f(v),\, g(z,\, v))\}$. Используем описанный алгоритм поиска НОУ$(w) = \sigma$:

Шаг 1. Структура литер идентична.

Шаг 2. $n = 0$, $w_0 = w$, $\sigma_0 = \varnothing$.

Шаг 3. $\#w_0 = 2 \neq 1$.

Шаг 4. $\Delta_0 = \{x, f(z)\}$.

Шаг 5. $\sigma_1 = \sigma_0\{f(z)/x\} = \{f(z)/x\}$, $w_1 = w_0\{f(z)/x\} = \{P(f(z), f(f(z)), y), P(f(z), f(v), g(z, v))\}$, $n = 1$.

Шаг 3. $\#w_1 = 2 \neq 1$.

Шаг 4. $\Delta_1 = \{f(z), v\}$.

Шаг 5. $\sigma_2 = \sigma_1\{f(z)/v\} = \{f(z)/x,\ f(z)/v\}$, $w_2 = w_1\{f(z)/v\} = \{P(f(z), f(f(z)), y), P(f(z), f(f(z)), g(z, f(z)))\}$, $n = 2$.

Шаг 3. $\#w_2 = 2 \neq 1$.

Шаг 4. $\Delta_2 = \{y, g(z, f(z))\}$.

Шаг 5. $\sigma_3 = \sigma_2\{g(z,\, f(z))/y\} = \{f(z)/x,\ f(z)/v,\ g(z,\, f(z))/y\}$, $w_3 = w_2\{g(z, f(z))/y\} = \{P(f(z), f(f(z)), g(z, f(z))), P(f(z), f(f(z)), g(z, f(z)))\}$, $n = 3$.

Шаг 3. $\#w_3 = 1$, НОУ$(w) = \sigma = \sigma_3 = \{f(z)/x, f(z)/v, g(z, f(z))/y\}$.

(У128) Без ограничения общности можно считать, что любые два дизъюнкта из клаузального множества S не имеют общих переменных, т. к. если это не так, то можно заменить в любом дизъюнкте любые переменные на новые, не входящие в S, и перейти от рассмотрения S к рассмотрению полученного таким образом S'. [Нужно доказать, что для дизъюнктов $C_1(x)$ и $C_2(x)$, из невыполнимости $\forall x(C_1(x)\ \&\ C_2(x))$ следует невыполнимость $\forall x \forall y(C_1(x)C_2(y))$. Действительно, если при некотором $x = a$ матрица первой формулы ложна, то будет ложной и матрица второй формулы при $x = a$ или $y = a$.]

Пусть $C_1 = C_1' \vee l_1$ и $C_2 = C_2' \vee \neg l_2$ — дизъюнкты, не имеющие общих переменных, а l_1 и l_2 — литеры. Пусть σ — НОУ для $\{l_1, l_2\}$. Тогда формула $(C_1' \vee C_2')\sigma$ называется **резольвентой** C_1 и C_2. Правило вывода резольвенты $\dfrac{C_1' \vee l_1 \quad C_2' \vee \neg l_2}{(C_1' \vee C_2')\sigma}$ называется **правилом резолюции**. Применение правила резолюции к дизъюнктам l_1 и $\neg l_2$ (l_1 и l_2 — унифицируемы) приводит к выводу пустого дизъюнкта, $\square$.

Правило вывода $\dfrac{C \vee l_1 \vee l_2}{(C \vee l_1)\sigma}$ называется **правилом склейки**, а $(C \vee l_1)\sigma$ — **склейкой** $C \vee l_1 \vee l_2$.

(П) $C = P(x) \vee P(f(y)) \vee \neg Q(x)$, $\sigma = \{f(y)/x\}$, следовательно, $C\sigma = P(f(y)) \vee \neg Q(f(y))$.

(У129) Резольвента дизъюнктов C_1 и C_2 является их логическим следствием, т.е. $C_1, C_2 \vDash (C_1' \vee C_2')\sigma$. []

(У130) Склейка дизъюнкта является его логическим следствием, т.е. $C \vee l_1 \vee l_2 \vDash (C \vee l_1)\sigma$. [Очевидно]

(П) Пусть $S = \{P(x) \vee P(f(y)) \vee P(g(y)), \neg P(f(g(a))) \vee Q(b)\}$. Строим резолютивный вывод. Две первых формулы в выводе берутся последовательно из S и далее получается

3) НОУ$(P(x), P(f(y))) = \{f(y)/x\} = \sigma$
   (1σ) $P(f(y)) \vee P(g(y))$ — склейка 1-го дизъюнкта;
4) НОУ$(P(f(y)), P(f(g(a)))) = \{g(a)/y\} = \lambda$
   $(2, 3, \lambda)$ $P(g(g(a))) \vee Q(b)$ — по правилу резолюции (резольвента — логическое следствие S).

(У131) **Теорема о полноте метода резолюций.** Клаузальное множество S невыполнимо тогда и только тогда, когда существует резолютивный вывод $\square$ из S. []

Т. о. если клаузальное множество S невыполнимо, то, порождая все резольвенты из S, за конечное число шагов получим пустой дизъюнкт.

(У132) Если клаузальное множество S выполнимо, то, порождая резольвенты из S, получим, что либо, начиная с некоторого шага, резольвенты начнут повторяться, либо процесс бесконечного порождения новых резольвент. Первый случай позволяет выявить необщезначимость исходной формулы, второй — не дает никакой информации. []

(П) Пусть $S = \{Q(a), \neg Q(x) \vee Q(f(x))\}$. Строим резолютивный вывод. Две первые формулы в выводе берутся последовательно из S и далее получается

3) $(1, 2, \{a/x\})$ $Q(f(a))$;
4) $(2, 3, \{f(a)/x\})$ $Q(f(f(a)))$;

и т.д. до бесконечности.

(У133) **Теорема Чёрча.** Не существует алгоритма, дающего по произвольной формуле логики предикатов 1-го порядка A ответ на вопрос: "Общезначима ли A?" В формулировке теоремы используется точное понятие алгоритма, совпадающее, например, с определением машины Тьюринга. []

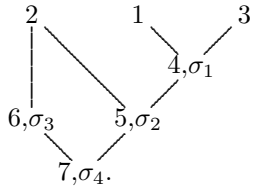
(У134) Для формул, содержащих только высказывания и свойства (одноместные предикаты), существует алгоритм, позволяющий выявить (не)общезначимость каждой из них. []

(П) Проверим методом резолюций общезначимость формулы $A = \Phi_1 \;\&\; \Phi_2 \Rightarrow \Phi_0$, где $\Phi_1 = \forall u \forall v \forall w (P(u, v) \;\&\; P(v, w) \Rightarrow Q(u, w))$, $\Phi_2 = \forall y P(f(y), y)$, $\Phi_0 = \exists x Q(x, a)$. Пусть, кроме того, задана следующая

интерпретация I: $P(x,y)$ означает "x отец y"; $Q(x,y)$ — "x дед y"; $f(x)$ — функция, возвращающая имя отца по имени x его сына; символу константы a сопоставим имя "Иван". Таким образом, нужно доказать, что из формул Φ_1 и Φ_2 следует, что у Ивана есть дедушка. Сначала проверяем является ли формула A предложением — является, т. к. в ней все переменные связаны. Построим отрицание A $A_0 = \neg A = \neg(\Phi_1 \,\&\, \Phi_2 \Rightarrow \Phi_0) \sim \neg(\neg\Phi_1 \vee \neg\Phi_2 \vee \Phi_0) \sim \Phi_1 \,\&\, \Phi_2 \,\&\, \neg\Phi_0 \sim (\forall u\forall v\forall w(P(u,v) \,\&\, P(v,w) \Rightarrow Q(u,w))) \,\&\, (\forall y P(f(y),y)) \,\&\, \neg(\exists x Q(x,a)) \sim (\forall u\forall v\forall w(\neg P(u, v) \vee \neg P(v,\ w) \vee Q(u,\ w))) \,\&\, \forall y P(f(y),\ y) \,\&\, \forall x\neg Q(x,\ a)$. Вынесем в префикс все кванторы $A_1 = \forall u\forall v\forall w\forall y\forall x((\neg P(u,\ v) \vee \neg P(v,\ w) \vee Q(u, w)) \,\&\, P(f(y), y) \,\&\, \neg Q(x,a))$ — это клаузальная форма формулы $\neg A$, следовательно, $S = \{(\neg P(u,v)\vee\neg P(v,w)\vee Q(u,w)), P(f(y),y), \neg Q(x,a)\}$. Строим резолютивный вывод. Три первых формулы в выводе берутся последовательно из S и далее получается

4) $(1,3,\sigma_1 = \text{НОУ}(Q(u,w),Q(x,a)) = \{x/u, a/w\})$ $\neg P(x,v) \vee \neg P(v,a)$;
5) $(2,4,\sigma_2 = \text{НОУ}(P(f(y),y),P(x,v)) = \{f(y)/x, y/v\})$ $\neg P(y,a)$;
6) $(2,\sigma_3 = \{z/y\})$ $P(f(z),z)$ — замена переменной y во 2-м дизъюнкте на z;
7) $(5,6,\sigma_4 = \text{НОУ}(P(f(z),z),P(y,a)) = \{f(a)/y, a/z\})$ $\square$.

Этот вывод можно изобразить следующим образом:



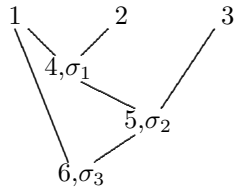
Доказано, что формула A общезначима. Кроме того, найдена формула для нахождения имени дедушки, x, т. к. в результате подстановок получается, что $x\sigma_1\sigma_2\sigma_4 = f(f(a))$. Т.о., приведенное доказательство — конструктивное.

(П) Проверим методом резолюций общезначимость формулы $A = \Phi_1 \,\&\, \Phi_2 \Rightarrow \Phi_0$, где $\Phi_1 = \forall y(R(y) \Rightarrow T(0))$, $\Phi_2 = \forall z\neg R(z) \Rightarrow T(1)$, $\Phi_0 = \exists x T(x)$. Пусть также задана следующая интерпретация I: $R(x)$ означает "x — живое существо, обитающее на Марсе"; $T(x)$ означает "на Марсе есть жизнь, если $x = 0$, и на Марсе жизни нет, если $x \neq 0$". Таким образом, нужно выяснить, следует ли из утверждений Φ_1 и Φ_2 ответ на вопрос: "Есть ли жизнь на Марсе?" Опять сначала проверяем является ли формула A предложением — является. Построим отрицание A $A_0 = \neg A = \neg(\Phi_1 \,\&\, \Phi_2 \Rightarrow \Phi_0) \sim \neg(\neg\Phi_1 \vee \neg\Phi_2 \vee \Phi_0) \sim \Phi_1 \,\&\, \Phi_2 \,\&\, \neg\Phi_0 \sim (\forall y(R(y) \Rightarrow T(0))) \,\&\, (\forall z\neg R(z) \Rightarrow T(1)) \,\&\, \neg(\exists x T(x)) \sim (\forall y(\neg R(y) \vee T(0))) \,\&\, (\exists z(R(z) \vee T(1))) \,\&\, (\forall x\neg T(x))$. Вынесем в префикс

все кванторы $A_1 = \exists z \forall y \forall x((\neg R(y) \vee T(0)) \ \& \ (R(z) \vee T(1)) \ \& \ \neg T(x))$. Пусть a — сколемовская константа. Тогда сколемовской нормальной формой $\neg A$ будет $A_2 = \forall y \forall x((\neg R(y) \vee T(0)) \ \& \ (R(a) \vee T(1)) \ \& \ \neg T(x))$, которая есть и клаузальная форма $\neg A$. Следовательно, $S = \{\neg T(x), \neg R(y) \vee T(0), R(a) \vee T(1)\}$. Построим резолютивный вывод $\square$. Три первых формулы в выводе берутся последовательно из S и далее получается

4) $(1, 2, \sigma_1 = \text{НОУ}(T(x), T(0)) = \{0/x\})\ \neg R(y)$;
5) $(3, 4, \sigma_2 = \text{НОУ}(R(y), R(a)) = \{a/y\})\ T(1)$;
6) $(1, 5, \sigma_3 = \text{НОУ}(T(x), T(1)) = \{1/x\})\ \square$.

Этот вывод можно изобразить следующим образом:



Получено, что формула A общезначима, но доказательство неконструктивно, т.к. в x подставлялись и 0 и 1 ($x\sigma_1\sigma_2\sigma_3 = 0$, $x\sigma_3 = 1$). Итак, ответа на вопрос: “Есть ли жизнь на Марсе?” — получить не удалось...

Предложения Хорна

Рассмотрим общий вид дизъюнкта в клаузальном множестве — $\neg G_1 \vee \cdots \vee \neg G_k \vee F_1 \vee \cdots \vee F_l$ или $G_1 \ \& \ \cdots \ \& \ G_k \Rightarrow F_1 \vee \cdots \vee F_l$. Возможны следующие случаи:

1. $k = 0$, $l = 1$. Этот случай соответствует дизъюнкту, состоящему из единственного предиката. Если термы в предикате — константы, то ему соответствует факт, например, предикат ПРОСТОЕ(7) означает, что 7 — простое число. Если термы в предикате содержат переменные, то ему соответствует общезначимое представление для группы фактов, например, предикат ТЕЧИЗМ(x) означает, что все течет и все изменяется. Предложения этого типа записывают в виде $\Rightarrow F$;
2. $k > 0$, $l = 0$. Этот случай соответствует дизъюнкту вида $G_1 \ \& \ \cdots \ \& \ G_k \Rightarrow$. Обычно он используется для описания вопроса;
3. $k > 0, l = 1$. Этот случай соответствует знанию в форме ЕСЛИ-ТО и записывается в виде $G_1 \ \& \ \cdots \ \& \ G_k \Rightarrow F$;
4. $k = 0$, $l > 1$. Этот случай соответствует дизъюнкту вида $\Rightarrow F_1 \vee \cdots \vee F_l$. Он используется для представления фактов, содержащих нечеткости. Нечеткость обусловлена тем, что неясно какой из предикатов $F_1, \cdots, F_l$ — истинен;

5. $k > 0$, $l > 1$. Например, к этому типу относится предложение вида РОДИТЕЛЬ$(x, y) \Rightarrow$ ОТЕЦ$(x, y) \vee$ МАТЬ(x, y), означающее, что из того, что x — родитель y следует, что x либо отец, либо мать y.

Предложения (дизъюнкты) типа 1, 2 и 3 называются **предложениями Хорна** или **хорновскими дизъюнктами**. Предложения типа 1 называют **фактами**, типа 2 — **запросами**, а типа 3 — **правилами**.

Логическое программирования

(П) Пусть задана система уравнений. При традиционном, командном подходе написать программу для ее решения можно, только зная метод решения подобных задач, а также способ перевода этого метода в машинную форму. При декларативном подходе для решения этой задачи нужно лишь формально описать ее, а машина автоматически выберет метод решения и решит задачу.

Одним из видов декларативного подхода к написанию программ является **логическое программирование**, при котором требующая решения проблема описывается на языке логики предикатов 1-го порядка, а вычислительная машина методом резолюций находит ее решение. Если использовать для написания логических программ язык логики предикатов 1-го порядка без всяких ограничений, то это приведет к переборному, т.е. непрактичному алгоритму поиска решений. Поэтому для написания логических программ используют только часть языка логики предикатов 1-го порядка, а именно язык предложений Хорна, для которого в начале 1970-х годов был разработан практичный метод поиска решений — метод линейных резолюций.

В языке Хорна для описания логических программ допустимы только формулы вида $\forall x_1 \dots \forall x_n (G_1 \,\&\, \dots \,\&\, G_k \Rightarrow F)$ и $\forall x_1 \dots \forall x_n F$, а для запросов к логическим программам — только формулы вида $\exists x_1 \dots \exists x_n (G_1 \,\&\, \dots \,\&\, G_k)$. При резолютивном выводе используют не сам запрос, а его отрицание. Все три вида формул не должны содержать свободных переменных и все G и F — атомарные формулы.

(П) Рассмотрим логическую программу, состоящую из одного факта ПРОСТОЕ(11). На запрос $\exists x$ПРОСТОЕ(x), означающий, что надо вывести все значения x, при которых запрос является следствием логической программы, будет выведено число 11. Можно усовершенствовать программу, добавив в нее еще один факт — ПРОСТОЕ(5). Рассмотренный запрос к этой программе приведет к выводу чисел 5 и 11.

Для записи логических программ используют специальный синтаксис. Правила записываются в виде $F :- G_1, \dots, G_k$, факты — в виде F, а запросы — $? - G_1, \dots, G_k$. Литера F в программном утверждении называется **заголовком**, а последовательность литер $G_1, \dots, G_k$ — **телом**. Запрос $? - G_1, \dots, G_k$ называется **целевым утверждени-**

ем (**целью**), а $G_1, \ldots, G_k$ — **подцелями**. Все переменные, входящие в запрос, называются **целевыми**.

Если дано конечное и непустое множество предложений Хорна типа 1 и 3, то **логической программой**, состоящей из них, называется множество всех этих предложений (это клаузальное множество, соответствующее формуле, состоящей из конъюнкции этих предложений). Логические программы обозначаются буквой P, а запросы к ним — буквой G. Факты и правила логической программы называются **программными утверждениями**. Логические программы часто называют **базами данных** или **знаний**.

Пусть P — логическая программа. Тогда ее **сигнатурой** называется множество всех констант, предикатных и функциональных символов из P. Обозначение S_P. Если в P нет констант, то добавим к S_P произвольную константу. Эрбрановский универсум для P будем обозначать H_P, а эрбрановский базис — B_P. В качестве H-интерпретации возьмем интерпретацию, являющеюся моделью P, которая задается множеством $J_P \subset B_P$ таким, что истинны те и только те основные атомы, которые входят в J_P.

Модель P, в которой истинны только предложения из P, называется **минимальной эрбрановой моделью** P и обозначается $\min J_P$. Такая модель всегда существует.

Выполнение логической программы P в ответ на запрос G — это конструктивное доказательство того, что $\vDash P \Rightarrow G$, и нахождение подстановки θ, убирающей все переменные из G, такой, что $\vDash P \Rightarrow G\theta$.

(П) Рассмотрим три формулы: $A_0 = \exists y Q(a, y)$, $A_1 = \forall x P(f(x), x)$ и $A_2 = \forall u \forall v \forall w (P(v, u)\ \&\ P(v, w) \Rightarrow Q(u, w))$. Пусть нужно проверить формулу $A_1\ \&\ A_2 \Rightarrow A_0$ на общезначимость. Формулы A_1 и A_2 не подходят для записи запроса, а формула A_0 не может быть программным утверждением. Составим логическую программу из двух утверждений:

1) $P(f(x), x)$
2) $Q(u, w) :- P(v, u), P(v, w)$ $\%Q(u, w) \vee \neg P(v, u) \vee \neg P(v, w)$.

Запросом к ней будет

0) $? - Q(a, y)$ $\%\neg Q(a, y)$.

Построим вывод $\square$:

3) $(0, 2, \sigma_1 = \text{НОУ}(Q(u, w), Q(a, y)) = \{w/y, a/u\})$ $? - P(v, a), P(v, w)$ $\%\neg P(v, a) \vee \neg P(v, w)$;
4) $(1, 3, \sigma_2 = \text{НОУ}(P(f(x), x), P(v, a)) = \{a/x, f(a)/v\})$ $? - P(f(a), w)$ $\%\neg P(f(a), w)$;
5) $(1, 4, \sigma_3 = \text{НОУ}(P(f(x), x), P(f(a), w)) = \{a/x, a/w\})$ $\square$.

Следовательно, $\vDash A_1\ \&\ A_2 \Rightarrow A_0$ при $y = a$, т. к. $y\sigma_1\sigma_2\sigma_3 = a$. Здесь $S_P = \{a,\ f,\ P,\ Q\}$, $H_P = \{a,\ f(a),\ f(f(a)),\ \ldots\}$, $B_P = \{P(a,\ a),\ Q(a,$

$a)$, $P(f(a), a), \ldots\}$. Определим теперь J_P. Если взять $J_P = \varnothing$, то все основные атомы будут ложны и, следовательно, первое утверждение из программы будет также всегда ложным, что противоречит определению модели. Если же взять $J_P = \{P(f(x), x) \mid x \in H_P\} \cup \{Q(x, y) \mid x, y \in H_P\}$, то J_P будет определять модель P. У P могут быть и другие модели.

Пусть P — логическая программа и G — запрос к ней, содержащий k переменных. Тогда если $k = 0$, то **правильным ответом** на запрос G к P является *Да* (или пустая подстановка), если $P \models G$, и *Нет*, если G не следует из P. Если же $k > 0$, то правильным ответом на G к P является подстановка θ такая, что $G\theta$ не содержит переменных и $P \models G\theta$. В случае, если такой подстановки не существует, то правильный ответ — *Нет*.

Метод линейных (SLD — Selector function, Linear, Definite clauses) резолюций

Правилом вычисления R называется функция, которая выделяет из целевого утверждения некоторую подцель.

Пусть G_i — целевое утверждение вида $? - A_1, \ldots, A_m, \ldots, A_k$, а C_{i+1} — программное утверждение вида $A : -B_1, \ldots, B_l$; C_{i+1} не имеет общих переменных с G_i. Тогда целевое утверждение G_{i+1} **выводится** из G_i и C_{i+1} при помощи подстановки θ_{i+1} по правилу R, если: 1) A_m выбирается из G_i по правилу R; 2) $\theta_{i+1} = \text{НОУ}(A, A_m)$; 3) G_{i+1} — это запрос вида $? - (A_1, \ldots, A_{m-1}, B_1, \ldots, B_l, A_{m+1}, \ldots, A_k)\theta_{i+1}$, т. е. резольвента C_{i+1} и G_i с НОУ θ_{i+1}.

Дизъюнкты C и C' называются **вариантами** друг друга, если каждый из них можно получить из другого переименованием переменных.

Пусть P — логическая программа, G — целевое утверждение, R — правило вычисления. Тогда **линейный резолютивный вывод** из $P \cup \{\neg G\}$ состоит из:

1. конечной или бесконечной последовательности целевых утверждений $G = G_0, G_1, \ldots$;
2. последовательности вариантов программных утверждений из P $C_1, C_2, \ldots$;
3. последовательности подстановок $\theta_1, \theta_2, \ldots$ — таких, что G_{i+1} выводится из G_i и C_{i+1} при помощи подстановки θ_{i+1} по правилу R и C_{i+1} не имеет общих переменных с $G_0, \ldots, G_i$.

Линейное резолютивное опровержение множества $P \cup \{\neg G\}$ с помощью правила R — это линейный резолютивный вывод из $P \cup \{\neg G\}$ пустого дизъюнкта.

(П) Рассмотрим логическую программу и запрос из предыдущего

примера и линейный резолютивный вывод $\square$ (правило вычисления — брать самый левый атом в запросе):

$$\begin{array}{lll} & & G_0 = G = Q(a,y) \\ C_1 = Q(u,w) : -P(v,u), P(v,w) & \theta_1 = \{w/y, a/u\} & G_1 = P(v,a), P(v,w) \\ C_2 = P(f(x),x) & \theta_2 = \{f(a)/v, a/x\} & G_2 = P(f(a),w) \\ C_3 = P(f(x),x) & \theta_3 = \{a/x, a/w\} & G_3 = \square \end{array}$$

$$\theta_1\theta_2\theta_3 = \{a/y, a/u, f(a)/v, a/x, a/w\} \qquad \theta = \{a/y\}.$$

(П) Рассмотрим логическую программу.

дуга(петербург,москва). %1

путь(X,X). %2

путь(X,Z) :- дуга(X,Y), путь(Y,Z). %3

Она описывает поиск пути в графе. Составим запрос

?-путь(X,москва),

означающий: “Из какого пункта X можно попасть в Москву?”. Правило R заключается в том, что всегда будет браться первая подцель. Получим 1-й вывод:

0) G=G_0=путь(X,москва)

1) C_1=2{X1/X}=путь(X1,X1)
   θ_1=НОУ(путь(X,москва), путь(X1,X1))={москва/X, москва/X1}
   G_1=$\square$, θ={москва/X},

т.е. в Москву можно попасть из Москвы.

Построим другой вывод, т.к. при выборе C_1 была альтернатива:

1) C_1=3{X1/X}=путь(X1,Z):-дуга(X1,Y),путь(Y,Z)
   θ_1=НОУ(путь(X,москва), путь(X1,Z))={X/X1, москва/Z}
   G_1=дуга(X,Y),путь(Y,москва)

2) C_2=1=дуга(петербург,москва)
   θ_2=НОУ(дуга(X,Y),дуга(петербург,москва))=
   ={москва/Y,петербург/X}
   G_2=путь(москва,москва)

3) C_3=2{X1/X}=путь(X1,X1)
   θ_3=НОУ(путь(москва,москва), путь(X1,X1))={москва/X1}
   G_3=$\square$, θ={петербург/X},

т.е. в Москву можно попасть из Петербурга.

При выборе C_3 опять была альтернатива, поэтому:

3) C_3=3{X1/X, Y1/Y}=путь(X1,Z):-дуга(X1,Y1),путь(Y1,Z)
   θ_3=НОУ(путь(москва,москва),путь(X1,Z))=
   ={москва/X1,москва/Z}
   G_3=дуга(москва,Y1),путь(Y1,москва)

4) $\blacksquare$ — неуспех

Дальнейший вывод невозможен, т.к. литера дуга(москва,Y1) не может быть унифицирована ни с одним заголовком правила из P. Итак, в результате работы этой программы получим, что в Москву можно попасть из Москвы или из Петербурга. Если изменить R и брать последнюю подцель, то в 2-м выводе получим:

1) G_1=дуга(X,Y),путь(Y,москва)
2) C_2=3{X1/X, Y1/Y}=путь(X1,Z):-дуга(X1,Y1),путь(Y1,Z)
   θ_2=НОУ(путь(Y,москва), путь(X1,Z))={Y/X1, москва/Z}
   G_2=дуга(X,Y),дуга(Y,Y1),путь(Y1,москва)
   и т.д. до бесконечности.

Семантика логических программ

Рассмотрение логической программы как описание некоторого мира при помощи предложений Хорна называется ее **декларативным** истолкованием (**декларативной моделью** или **семантикой**). Ответы на запрос при таком способе истолкования получаются путем применения логических правил вывода. При этом, очевидно, что порядок следования утверждений в логической программе не имеет значения.

Подцель G **сопоставима** с программным утверждением C, если для заголовка C и G существует НОУ.

При **процедурном** истолковании логической программы (**процедурной модели**):

1. Множество программных утверждений с одним и тем же предикатным символом в заголовке трактуется как описание одной **процедуры**;
2. Подцель запроса с именем того же предиката, что и в заголовке процедуры, трактуется как **вызов** этой процедуры;
3. Для того чтобы запрос оказался успешным, необходимо, чтобы вызываемые в нем процедуры были успешно выполнены;
4. Вызов процедуры считается **успешным**, если удалось унифицировать параметры вызова процедуры и параметры заголовка одного из вариантов утверждения вызываемой процедуры. Результатом вызова является замена произведенного вызова на тело выбранного утверждения и применение ко всей цели использованного НОУ. Вызов процедуры успешен тогда и только тогда, когда существует сопоставимое с ней программное утверждение;
5. Важен порядок вызова процедур в запросе и порядок расположения утверждений каждой процедуры.

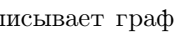
   (П) Рассмотрим предложение логической программы.

   начальник(Фамилия,Оклад) :- служащий(Фамилия,Оклад),
   Оклад>70000.

При декларативном истолковании ему соответствует фраза: “Человек является начальником, если он служащий и его оклад более 70000”. При процедурном истолковании — фраза: “Один из способов обнаружить начальника — это: 1) отыскать служащего; 2) проверить превышает ли его оклад 70000”.

(П) Рассмотрим логическую программу:

```
дуга(b,c). %1
дуга(a,b). %2
дуга(c,b). %3
путь(X,X). %4
путь(X,Z) :- дуга(X,Y), путь(Y,Z). %5
```

Она описывает граф 

и механизм поиска пути на нем. Пусть надо найти ответ на запрос

```
?-путь(X,c),путь(c,X).
```

Решим задачу в процедурной модели.

1. ?-дуга(X,Y),путь(Y,c),путь(c,X)	%5{X1/X}, {X/X1,c/Z}
2. ?-путь(c,c),путь(c,b)	%1, {b/X,c/Y}
3. ?-путь(c,b)	%4{X1/X}, {c/X1}
4. ?-дуга(c,Y1),путь(Y1,b)	%5{X1/X,Y1/Y}, {c/X1,b/Z}
5. ?-путь(b,b)	%3, {b/Y1}
6. □	%4{X1/X}, {b/X1}

Ответ: X=b.

Пусть P — логическая программа, G — запрос к ней, R — правило вычислений, $\theta_1, \ldots, \theta_n$ — последовательность подстановок линейного резолютивного опровержения $P \cup \{\neg G\}$. Тогда **R-вычислимым ответом** на запрос G к P называется подстановка θ, которая получается из композиции подстановок $\theta_1, \ldots, \theta_n$ удалением всех элементов вида t/x, где x — нецелевая переменная.

(У135) **Теорема о корректности метода линейных резолюций.** Пусть P — логическая программа, G — запрос к ней, R — правило вычислений. Тогда каждый R-вычислимый ответ на G к P является правильным ответом на G к P. []

Множеством успехов логической программы P называется множество тех основных атомов A из ее эрбранова базиса B_P, что существует линейное резолютивное опровержение множества $P \cup \{\neg A\}$, т.е. существует линейный резолютивный вывод □ из запроса A к P.

(У136) Множество успехов программы P совпадает с наименьшей эрбрановой моделью P, $\min J_P$. []

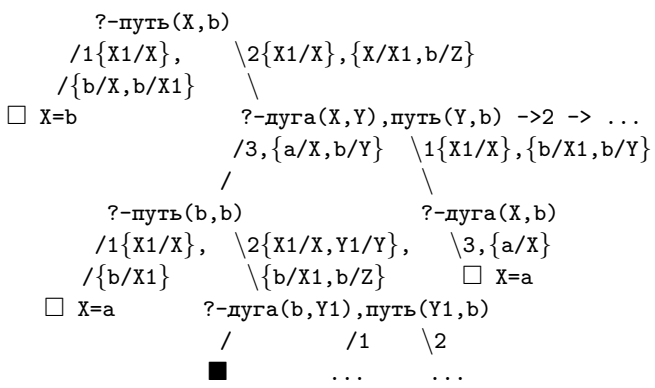
(У137) **Теорема о полноте метода линейных резолюций.** Пусть P — логическая программа, G — целевое утверждение, θ — правильный ответ на G к P. Тогда существует правило вычислений R

и R-вычислимый ответ σ на запрос G к P такой, что для некоторой подстановки γ $\theta = \sigma\gamma$. []

Полное дерево вычислений — это множество всех линейных резолютивных выводов при помощи всех возможных правил вычисления.

(П) Построим полное дерево вычислений для следующей программы.

```
путь(X,X). %1
путь(X,Z) :- дуга(X,Y), путь(Y,Z). %2
дуга(a,b). %3
```



Множество R-успехов программы P есть множество всех атомов $A \in B_P$ таких, что $P \cup \{\neg A\}$ имеет линейное резолютивное опровержение с R.

(У138) Для любых логических программ P и правил вычисления R множество R-успехов совпадает с наименьшей эрбрановской моделью P. []

(У139) **Теорема о сильной полноте метода линейных резолюций** Пусть P — логическая программа, G — целевое утверждение, R — правило вычислений. Тогда для любого правильного ответа θ на запрос G к P существует R-вычислимый ответ σ на запрос G к P и $\theta = \sigma\gamma$ для некоторой подстановки γ. []

Линейное резолютивное дерево для P, G и R:

1. вершины (узлы) — целевые утверждения;
2. корень G;
3. если $? - A_1, \ldots, A_m, \ldots, A_k$ — узел и A_m выбирается по R, то для любого варианта $A : -B_1, \ldots, B_s \in P$ такого, что существует $\theta = \text{НОУ}(A, A_m)$, под означенным узлом лежит узел $? - (A_1, \ldots, A_{m-1}, B_1, \ldots, B_s, A_{m+1}, \ldots, A_k)\theta$.

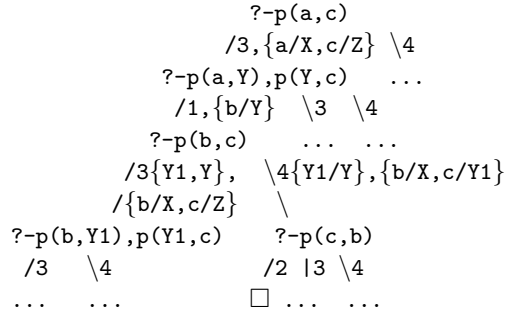
Правила, определяющие стратегию вычисления логической программы:

1. правило вычисления (выбор вызова);
2. правило поиска (в глубину с возвратом, в ширину);
3. правило выбора применяемых программных утверждений.

В языке программирования ПРОЛОГ правило вычисления — это активизация самого левого атома в запросе. Поиск решений производится в глубину с возвратом. Утверждения выбираются в порядке их следования в программе.

(П) Следующая логическая программа содержит в себе информацию, необходимую и достаточную для получения правильного ответа на заданный вопрос, но по правилам, определяющим стратегию вычисления пролог-программы, его получить невозможно.

```
p(a,b). %1
p(c,b). %2
p(X,Z) :- p(X,Y), p(Y,Z). %3
p(X,Y) :- p(Y,X). %4
```



Смена правила вычисления или порядка следования программных предложений ничего не изменит — пустой дизъюнкт не выведется. Здесь виден недостаток поиска в глубину, но поиск в ширину требует очень много памяти.

В стандарте ISO языка пролог в конце программных и целевых утверждений нужно ставить точку, комментарии заключаются между символом процента и концом строки. Переменные должны начинаться с заглавной буквы, константы и предикаты — со строчной.

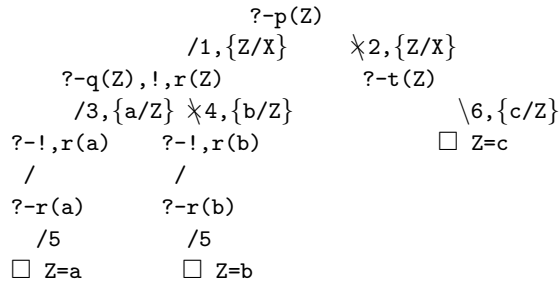
Стандартные средства пролога

Предикат **отсечения** (!) отсекает ветви в линейном резолютивном дереве. Он — всегда истинен и поэтому не имеет смысла в декларативной семантике. В процедурной семантике после его выполнения прекращается поиск соответствия для подцели, сопоставленной заголовку правила, содержащего в себе !. Отсечение играет роль ловушки

и припятствует возврату к целям, расположенным левее него, включая заголовок.

(П) Рассмотрим логическую программу, содержащую в себе отсечение.

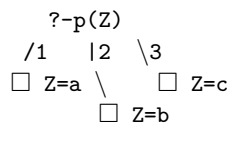
p(X) :- q(X), !, r(X). %1
p(X) :- t(X). %2
q(a). %3
q(b). %4
r(X). %5
t(c). %6



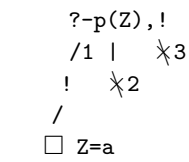
Ответ: Z=a.

Использование ! в конце запроса гарантирует получение только одного ответа.

(П) p(a). %1
p(b). %2
p(c). %3



Ответ: Z=a, Z=b, Z=c.



Ответ: Z=a.

Если в запросе нет переменных, то поиск решения продолжается только до вывода первого пустого дизъюнкта, т.е. запрос без переменных можно рассматривать как запрос, содержащий в конце отсечение.

Предикат fail — всегда ложен. Его можно осмысленно использовать только в процедурной семантике. Факт fail не должен присутствовать в пролог-программе. Предикат fail часто используется после ! в конце правил.

(П) Нужно выразить на прологе фразу: “Мэри любит всех животных, кроме змей”.

love(mary,X) :- snake(X), !, fail. %1

love(mary,X) :- animal(X). %2

Дополним программу фактами.

snake(python). %3

snake(cobra). %4

animal(python). %5

animal(rabbit). %6

```
        ?-love(mary,python) %Любит ли Мэри питона?
        /1,{pyton/X}     ↯2
   ?-snake(python),!,fail
    /3
 ?-!,fail
     /
    ■
```

Ответ: Нет (No).

```
          ?-love(mary,rabbit) %Любит ли Мэри кролика?
         /1,{rabbit/X}      \2,{rabbit/X}
 ?-snake(rabbit),!,fail    ?-animal(rabbit)
      /                      \6
     ■                        □
```

Ответ: Да (Yes).

Спросим: “Кого любит Мэри?”

```
        ?-love(mary,X)
       /1{X1/X},{X/X1}  ↯2
   ?-snake(X),!,fail
   /3,{python/X}  ↯4
 ?-!,fail
     /
    ■
```

Ответ: никого (No solution).

Предикат **равно** (=) — истинен, если существует НОУ его аргументов. Кроме того, в случае успешной унификации переменным присваиваются соответствующие значения.

(П) ?-X=a → □ Ответ: X=a.

?-f(a,Y)=f(X,b) → □ Ответ: X=a, Y=b.

?-a+Y=X+7 → □ Ответ: X=a, Y=7.

?-f(a,b)=f(a,c) → ■ Ответ: Нет (No).

Предикат is отличается от предиката = тем, что унификация происходит после вычисления правой части, арифметического выражения.

(П) ?-X is 7+5 → □ Ответ: X=12.

?-12 is 7+5 → □ Ответ: Да (Yes).

?-X=7, X is 3+2 $-\{7/X\}\rightarrow$?-7 is 3+2 $\rightarrow \blacksquare$ Ответ: Нет (No).

Предикат true — всегда истинен.

ИЛИ в прологе

Если два программных утверждения имеют один и тот же заголовок, то их можно соединить в одну конструкцию, в которой заголовок останется тем же, а тела будут разделены знаком “точка с запятой” (записью операции *ИЛИ* в прологе).

(П) Следующие два правила:

p(X,Y) :- p(X,Z), p(Z,Y).

p(X,Y) :- p(Y,X).

— можно записать как одно

p(X,Y) :- p(X,Z), p(Z,Y); p(Y,X).

Предикат true можно осмысленно использовать только совместно с *ИЛИ*, в сокращениях.

(П) Логическую программу

p(X) :- t(X).

p(X).

p(X) :- r(X).

можно записать в следующей форме

p(X) :- t(X); true; r(X).

Отрицание в прологе

В телах правил и запросах допустимо использовать отрицание. Для обозначения отрицания будет использоваться знак ~ (тильда). В пролог-системах отрицание иногда обозначается словом not. Введение отрицания приводит к нарушению хорновости и, в частности, к потере конструктивности в выводе.

(П†) p(0) :- q. %1 $\neg q \vee p(0)$

p(1) :- ~q. %2 $q \vee p(1)$ — нехорновский дизъюнкт

?-p(X). %3 $\neg p(X)$

Используя метод резолюций к клаузальному множеству из этих 3-х дизъюнктов, получим следующий вывод:

4) $(1, 3, \{0/X\})\ \neg q$

5) $(2, 3, \{1/X\})\ q$

6) $(4, 5)\ \square$

В X подставляется как 0, так и 1. Получен неконструктивный вывод.

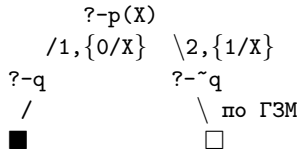
Введение отрицания, если понимать его буквально, приводит к невозможности строить эффективные линейные резолютивные выводы. Но в прологе отрицание понимается не буквально, а в контексте **гипотезы о замкнутости мира** (ГЗМ), которая заключается в том, что все, что не следует из программы — ложно.

(П) male(john).

male(adam).
female(mary).
female(julia).
?-˜male(mary). %Yes

Из программы не следует, что Мэри — немужчина, т. е. $P \not\models \neg\text{male(mary)}$, но т.к. $P \not\models \text{male(mary)}$, то по ГЗМ ответ будет *Да*.

(П) Используя ГЗМ в П†, получим конструктивное решение X=1:



ГЗМ можно записать в виде правила вывода $\dfrac{P \not\models A}{\neg A}$, в котором очень нетривиальная посылка. ГЗМ приводит к нарушению монотонности отношений $\vdash$ и $\models$, т.е. свойств $\Gamma \vdash A \Rightarrow \Gamma, B \vdash A$ и $\Gamma \models A \Rightarrow \Gamma, B \models A$ для любого B. Кроме того, ГЗМ носит слишком общий характер, что не допускает его реализации в прологе: ГЗМ можно практически использовать только, если линейное резолютивное дерево конечно.

Вместо ГЗМ в прологе используется правило вывода **отрицание как неуспех** (ОКН): “Если линейное резолютивное дерево на запрос G к программе P конечно и не содержит успехов, то из P следует $\neg G$”. ОКН более слабое, чем ГЗМ.

(П) p(X) :- p(f(X)).
?-˜p(a).

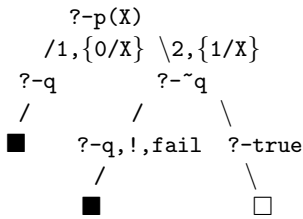
Нет ответа по ОКН, ответ *Да* по ГЗМ.

Согласно ОКН отрицание можно описать правилом

˜p :- p, !, fail; true.

Если очередная подцель начинается с отрицания, то происходит замена согласно выписанному правилу.

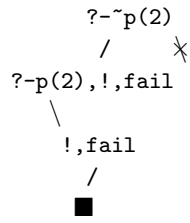
(П) Используем ОКН в П†:



Ответ: X=1, но p(1) логически не следует из заданной программы.

При использовании отрицания при выводе ответом может быть только *Да* или *Нет*.

(П) p(2).



Ответ: Нет.

Динамическое изменение логических программ

Всегда истинные предикаты asserta(C) и assertz(C) добавляют к программе (базе знаний) новое утверждение C (assert — утверждать): asserta — к началу программы, assertz — к концу.

Предикат retract(C) удаляет из программы утверждение, сопоставимое поатомарно с C (retract — отрекаться). Он истинен тогда и только тогда, когда поатомарно сопоставимое с C утверждение действительно существует в программе.

Предикат retractall(H) удаляет из программы все утверждения, заголовок которых сопоставим с H. Он всегда истинен.

(П) f(a) :- assertz(g(b)). %1
f(b) :- assertz((g(a) :- g(b))). %2
h(c) :- retract(g(b)). %3
h(d) :- retract((g(a) :- g(b))). %4
m(e) :- retractall(g(X)). %5
?- f(X), g(a). % X=b
?- m(X), g(a). % Нет
?- h(c). % Нет
?- f(X), g(b). % X=a, X=b
?- h(d), g(a). % Нет

Анонимные переменные

Это переменные, именуемые символом подчеркивания. Они попарно различны и их конечное значение не выводится в ответ.

(П) На запрос ?-p(_,_) к логической программе p(a,b) будет получен ответ *Да*; на запрос ?-p(X,X) — *Нет*; ?-p(X,Y) — X=a, Y=b; ?-p(_,X) — X=b.

Вычислимость

Машина Тьюринга — это математическая модель идеализированного вычислительного устройства, которое состоит из:

1. бесконечной ленты, разделенной на ячейки, в каждую из которых записан один из символов из некоторого конечного множества $A = \{a_0, a_1, \ldots, a_n\}$, называемого **внешним алфавитом**. Среди символов внешнего алфавита выделяется специальный пустой символ a_0, который обычно есть 0;
2. внутренней памяти, которая может находиться в одном из конечного числа состояний, которые обозначаются символами из некоторого конечного множества $Q = \{q_0, q_1, q_2, \ldots, q_m\}$, $Q \cap A = \varnothing$, называемого **внутренним алфавитом**. Среди состояний выделяют начальное — q_1 и конечное — q_0;
3. управляющей головки, которая может перемещаться вдоль ленты таким образом, что в каждый момент времени она находится над определенной ячейкой ленты.

В зависимости от внутреннего состояния и от символа под головкой, машина Тьюринга переходит в новое внутреннее состояние, записывает символ в ячейку под головкой и перемещает головку. Головку можно сдвигать только на одну ячейку влево или вправо или оставлять на месте.

Программа машины Тьюринга — это конечная последовательность пятисимвольных команд вида $q_i a_j \to q_k a_l s$. Команда $q_i a_j \to q_k a_l s$ означает, что если машина находится в состоянии q_i и символ под головкой — a_j, то она переходит в состояние q_k, пишет в текущую ячейку символ a_l и головка совершает движение s, где $s \in \{S, L, R\}$, где L означает сдвиг влево, R — вправо и S — стояние на месте. При старте программы машина Тьюринга находится в начальном состоянии q_1, на ленте имеется только конечное число символов, отличных от пустого, и головка устанавливается либо на первый непустой, либо, если таких нет, — на произвольный символ. Если в процессе вычислений машина переходит в конечное состояние q_0, то это приводит к ее остановке.

Машина Тьюринга M (математическое понятие) — это совокупность трех компонент $\langle A, Q, \Pi \rangle$, где A — внешний алфавит, Q — внутренний алфавит и Π — программа.

Слово на ленте — это последовательность символов на ленте, заключенных между двумя крайними непустыми.

Далее в качестве A будем рассматривать только множество $\{0, 1\}$. Для представления числа n будет использоваться слово из n единиц.

Если машина Тьюринга, начав работу с некоторым словом на ленте, перейдет в заключительное состояние, то она называется **применимой** к этому слову. **Результатом** ее работы считается слово, оставшееся на ленте, после остановки машины. Если же машина Тьюринга, начав работу с некоторым словом, никогда не перейдет в заключитель-

ное состояние, то она является **не применимой** к этому слову.

(П) $q_1 0 \to q_0 1 S$

$q_1 1 \to q_1 1 R$

Эта машина Тьюринга, начав работу на ленте, на которой записаны подряд n единиц ($n \geqslant 0$), добавит еще одну единицу к этому ряду. Она реализует функцию $f(x) = x + 1$.

(П) $q_1 0 \to q_0 0 S$

$q_1 1 \to q_1 0 R$

Эта машина Тьюринга убирает все идущие подряд единицы с ленты, т.е. она реализует функцию 0, $f(x) = 0$.

Тезис Тьюринга. Всякий интуитивный алгоритм может быть реализован с помощью некоторой машины Тьюринга. [Этот тезис нельзя доказать, т.к. неясен точный смысл понятия "интуитивный алгоритм", но никто еще не придумал что-либо похожего на алгоритм, который было бы нельзя реализовать на машине Тьюринга. Поэтому считается, что точное математическое понятие алгоритма и машина Тьюринга — это одно и то же. Тезис Тьюринга эквивалентен тезису Черча.]

Функция называется **вычислимой**, если существует алгоритм, ее вычисляющий. Под алгоритмом понимается машина Тьюринга.

(У140) Используя язык пролог, можно вычислить любую вычислимую функцию. [Конструктивное — реализацией машины Тьюринга на прологе.]

Утверждения подобные этому доказаны или могут быть доказаны для практически всех языков программирования.

Номером $N(M)$ **машины Тьюринга** M с программой из r команд вида $q_{i_n} a_{j_n} \to q_{k_n} a_{l_n} s_{m_n}$, $1 \leqslant n \leqslant r$, $s_0 = S$, $s_1 = L$, $s_2 = R$ назовем число

$$N(M) = p_1^{i_1} p_2^{j_1} p_3^{k_1} p_4^{l_1} p_5^{m_1} p_6^{i_2} p_7^{j_2} p_8^{k_2} p_9^{l_2} p_{10}^{m_2} \cdots p_{5r-4}^{i_r} p_{5r-3}^{j_r} p_{5r-2}^{k_r} p_{5r-1}^{l_r} p_{5r}^{m_r},$$

где $p_1, \ldots, p_n, \ldots$ — это последовательность простых чисел, т.е. 2, 3, 5, 7, 11, ...

(П) Рассмотренная ранее машина, реализующая функцию $f(x) = 0$, имеет номер $2^1 3^0 5^0 7^0 11^0 13^1 17^1 19^1 23^0 29^2 = 2 \times 13 \times 17 \times 19 \times 29 \times 29 = 7,062,718$

Машина M, применимая к числу $N(M)$, называется **самоприменимой**. Машина M, не применимая к числу $N(M)$, называется **несамоприменимой**.

(П) Машина из предыдущего примера самоприменима, а машина, в программе которой нет ни одной команды перехода в заключительное состояние, несамоприменима.

(У141) Не существует алгоритма, который по любой машине Тьюринга устанавливает, самоприменима она или нет. [Предположим, что такой алгоритм существует: результат его работы — 1, если машина самоприменима, и 0 в противном случае. Модифицируем этот алгоритм: пусть вместо результата 1 он переходит на бесконечный цикл. Теперь применим этот алгоритм (машину Тьюринга) к себе самому. Если он самоприменим, то машина перейдет в бесконечный цикл, что означает, что он несамоприменим. Если же он несамоприменим, то машина остановится, что означает, что он самоприменим. Полученное противоречие доказывает утверждение.]

Вычислительная сложность алгоритмов

Решение многих задач математики носит алгоритмический характер (пример, алгоритм Евклида поиска НОД). Бурно развивающийся ныне ее раздел "Численные методы" целиком построен на изучении таких задач. Хотя есть алгоритмически неразрешимые проблемы, например, проблема проверки общезначимости формул логики предикатов или самоприменимости алгоритмов, большинство математических задач может быть решено тем или иным алгоритмом.

Временная сложность алгоритма — это зависимость времени его работы от размеров входных данных. Размером входных данных будем считать длину соответствующей им последовательности из 0 и 1, т. е. количество бит, необходимых для их кодирования. Под временем здесь понимается функция, монотонно возрастающая после выполнения каждого шага алгоритма.

Полиномиальным алгоритмом (или алгоритмом полиномиальной временной сложности) называется алгоритм, у которого временная сложность $t(n)$, где n — это размер входных данных, связана соотношением $t(n) \leqslant P(n)$ для всех n, где $P(n)$ — многочлен (полином) от n.

Алгоритмы, для временной сложности которых не существует полиномиальной оценки, называются **экспоненциальными**.

(П) Простейший способ решения задач дискретной математики — это перебор всех возможных вариантов, но это очень неэффективный способ, имеющий в общем случае экспоненциальную временную сложность и, следовательно, практически непригодный даже для самой мощной вычислительной техники.

(П) Решение системы линейных уравнений методом Гаусса имеет полиномиальную временную сложность, а методом Крамера, вычисляя определитель матрицы размерности $n \times n$ как сумму из $n!$ слагаемых, — экспоненциальную.

Задача считается **труднорешаемой**, если для нее не существует

разрешающего полиномиального алгоритма.

(П) Поиск всех циклов в графе — труднорешаемая задача.

(П) Рассмотрим размеры данных, которые могут обработать различные ЭВМ на алгоритмах с различной временной сложностью за единицу времени:

Функция временной сложности, $t(n)$	Современная ЭВМ	ЭВМ, в 100 раз более быстрая	ЭВМ, в 10000 раз более быстрая
n	N_1	$100 \times N_1$	$10000 \times N_1$
n^{20}	N_2	$1.26 \times N_2$	$1.59 \times N_2$
2^n	N_3	$N_3 + 6.64$	$N_3 + 13.29$

.

Если $N_1 = N_2 = N_3 = 1000$ бит, то увеличение быстродействия ЭВМ в 100 раз позволит обрабатывать 100000 бит, 1260 бит и лишь 1007 бит (рост $+0.7\%$) соответственно...

Сложность задачи — это сложность наилучшего алгоритма, известного для ее решения.

Задачи полиномиальной сложности образуют класс P. Задачи экспоненциальной сложности — класс E.

Обозначение $f(n) = O(g(n))$ означает, что для некоторой константы k и для любых n выполняется $|f(n)| \leqslant kg(n)$.

Типовые задачи класса P:

- Решение системы из n линейных уравнений. Сложность $O(n^2)$.
- Поиск эйлерова цикла в графе из m ребер. Сложность $O(m)$.
- Сортировка множества из n элементов. Сложность $O(n \log n)$.
- Поиск кратчайшего пути на графе из n вершин и m ребер между заданными вершинами. Сложность $O(mn)$.
- Проверка реализуемости графа из n вершин на плоскости. Сложность $O(n)$.
- Постройка минимального покрывающего дерева для заданного множества из n вершин. Другими словами, из полного нагруженного графа с n вершинами нужно выделить подграф-дерево с n вершинами и минимальным общим весом ребер. Практически такая задача может возникнуть, например, при проектировании сети водоснабжения. Сложность $O(n \log n)$.
- Разбиение множества из n вершин графа на подмножества связанных вершин. Иными словами, выделение из графа связных подграфов. Сложность $O(n^2)$.
- Вычисление максимального потока между двумя заданными вершинами нагруженного графа с n вершинами. Весовая функция определяет пропускную способность каждого ребра. Сложность $O(n^3)$.

- Линейное программирование. Наиболее широко используемый ныне симплекс-метод принадлежит классу E. В 1979 г. Хачианом предложен метод класса P. Сложность $O(n^3)$, где n — это число неизвестных в системе линейных уравнений.

Некоторые задачи класса E:

- Выделения в графе всех подграфов-деревьев.
- Распознавание правильных фраз на формальных языках с относительно простыми алфавитами и правилами построения фраз.

Типовые задачи, не входящие в класс P или E

- Решение систем уравнений в целых числах — решение диофантовых уравнений.
- Нахождение гамильтонова цикла в графе.
- Нахождение гамильтонова цикла минимального веса в нагруженном графе — задача коммивояжера.
- Проверка на общезначимость произвольной формулы логики высказываний.
- Оптимальная загрузка емкости — задача о рюкзаке.
- Выделение на графе наименьшего множества вершин таких, что каждое ребро графа будет инцидентно вершине из этого множества, — задача о вершинном покрытии.
- Выделение из конечного множества целых чисел подмножества, сумма элементов которого равна заданному числу.
- Оптимальный раскрой (бумага, ткань, стальной прокат).

Для всех перечисленных задач доказана их взаимная эквивалентность, т. е. нахождение полиномиального алгоритма для одной из них будет означать, что все они принадлежат классу P.

Класс NP

Недетерминированная машина Тьюринга отличается от (детерминированной) машины Тьюринга тем, что переход в новое состояние может определяться множеством состояний — каждый элемент множества создает копию существующей машины Тьюринга. Останов машины происходит тогда, когда одна из ее копий достигает конечного состояния.

(П) Если недетерминированная машина Тьюринга находится в состоянии q_i и под головкой на ленте находится 1, то команда $q_i 1 \to \{q_j, q_k\}1L$ создаст две машины Тьюринга, различающиеся только внутренними состояниями.

Класс задач NP (Nondeterministic Polynomial) составляют задачи, решаемые за полиномиальное время на недетерминированной машине Тьюринга.

(У142) $P \subset NP$. [Очевидно]

Задача называется **NP-сложной** или **NP-трудной** (NP-hard), если к ней за полиномиальное время может быть преобразована любая задача класса NP. NP-сложная задача не обязательно должна принадлежать классу NP.

Задача называется **NP-полной**, если она NP-сложная и принадлежит классу NP. Класс NP-полных задач обозначается NPC (NP-complete).

(У143) **Теорема Кука**. Задача проверки на общезначимость произвольной формулы логики высказываний является NP-сложной. [[10, стр. 214–218]]

Класс задач, дополнительных к задачам, принадлежащим классу NP, образует класс co-NP.

(У144) P $\subset$ co-NP. [Следует из того, что P = co-P.]

(У145) Из NP $\neq$ co-NP следует P $\neq$ NP. [[8, стр. 849]]

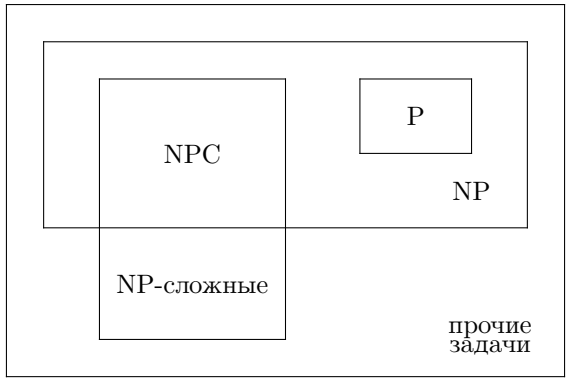
Типовые задачи класса NP:

- Существование гамильтонова цикла на графе. NP-полная.
- Задача о рюкзаке: для переменных x_i, принимающих значения из множества $\{0, 1\}$, найти решение уравнения $\sum_i a_i x_i = b$, где все a_i и b — заданные целые числа. NP-полная.
- Раскраска в 3 цвета вершин графа, так чтобы каждое ребро имело вершины разного цвета. NP-полная.
- Разбиение множества натуральных чисел на два непересекающихся подмножества так, чтобы сумма чисел в обоих подмножествах была одинакова.
- Выделение полного подграфа (клики) с заданным числом вершин на графе. NP-полная.
- Проверка на общезначимость произвольной формулы логики высказываний в виде КНФ, все дизъюнкты которой состоят из трех литер. NP-полная.
- Определения маршрута коммивояжера стоимости, меньшей заданного числа. NP-полная.
- Задача о выполнимости, заданной схемы из логических компонент. NP-полная.

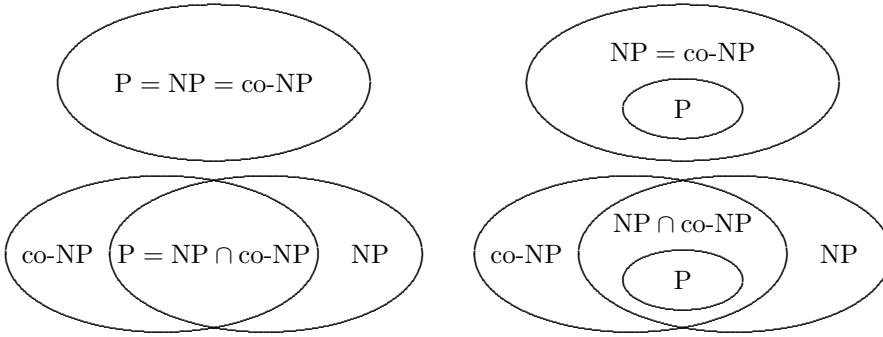
Задача нахождения оптимального маршрута коммивояжера, как и большинство задач на оптимизацию, не относится к классу NP.

Нерешенные проблемы теории класса NP

Совпадают ли P и NP? Это одна из основных проблем современной математики. Следующий рисунок иллюстрирует существующие представления о взаимосвязях компонент теории класса NP.



Как соотносятся между собой классы NP, co-NP и P? Возможны четыре варианта, представленные на следующем рисунке.



ЛИТЕРАТУРА

1. Derek F. Holt, Bettina Eick, Eamonn A. O'Brien *Handbook of computational group theory*/ Discrete mathematics and its applications, Series Editor Kenneth H. Rosen — Chapman & Hall/CRC Press, 2005. — 510 p.
2. Акимов О. Е. *Дискретная математика: логика, группы, графы*/ 2-е издание — М.: Лаборатория базовых знаний, 2001. — 376 с.
3. Биркгоф Г., Барти Т. *Современная прикладная алгебра* — М.: Мир, 1976. — 400 с.
4. Братко И. *Программирование на языке Пролог для искусственного интеллекта* — М.: Мир, 1990. — 560 с.
5. Верещагин Н. К., Шень А. *Лекции по математической логике и теории алгоритмов. Часть 1. Начала теории множеств* — М.: МЦНМО, 1999. — 128 с.
6. Верещагин Н. К., Шень А. *Лекции по математической логике и теории алгоритмов. Часть 2. Языки и исчисления* — М.: МЦНМО, 2000. — 288 с.

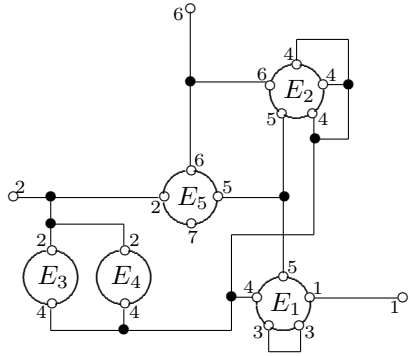
7. Ершов Ю. А., Палютин Е. А. *Математическая логика* — М.: Наука, 1987. — 336 с.
8. Кормен Т., Лейзерсон Ч., Ривест Р. *Алгоритмы построение и анализ* — М.: МЦНМО, 2000. — 960 с.
9. Кнут Д. Е. *Искусство программирования, том 3. Сортировка и поиск, 2-е изд.* — М.: Издательский дом "Вильямс", 2003. — 832 с.
10. Лорьер Жан-Луис *Системы искусственного интеллекта* — М.: Мир, 1991. — 568 с.
11. Маллас Дж. *Реляционный язык Пролог и его применение* — М.: Наука, 1990. — 464 с.
12. Мальцев А. И. *Алгоритмы и рекурсивные функции* — М.: Наука, 1986. — 368 с.
13. Нефедов В. Н., Осипова В. А. *Курс дискретной математики* — М.: МАИ, 1992. — 264 с.
14. Осуга С. *Обработка знаний* — М.: Мир, 1989. — 293 с.
15. Ноден П., Китте К. *Алгебраическая алгоритмика* — М.: Мир, 1999. — 720 с.
16. Хоггер К. *Введение в логическое программирование* — М.: Мир, 1988. — 348 с.
17. Стерлинг Л., Шапиро Э. *Искусство программирования на языке Пролог* — М.: Мир, 1990. — 235 с.
18. Тейз А., Грибомон П., Луи Ж. и др. *Логический подход к искусственному интеллекту* — М.: Мир, 1990. — 432 с.
19. Шикин Е.В. *Линейные пространства и отображения* — М.: Издательство Московского университета, 1987. — 311 с.
20. Яблонский С.В. *Введение в дискретную математику* — М.: Наука, 1979. — 272 с.

ПРИЛОЖЕНИЯ

Сети

Множество $M = \{a_1, a_2, \ldots\}$ и набор $N = \{E_0; E_1, E_2, \ldots\}$, в котором E_i есть набор элементов из M, называется **сетью** и обозначается $M(E_0; E_1, E_2, \ldots)$. Объекты множества M называются **вершинами**, а объекты из набора E_0 — **полюсами** сети. Сеть иногда называют **гиперграфом**.

(П) Пусть $M = \{1, 2, 3, 4, 5, 6, 7\}$, $N = \{E_0; E_1, E_2, E_3, E_4, E_5\}$, где $E_0 = \{1, 2, 6\}$, $E_1 = \{1, 3, 3, 4, 5\}$, $E_2 = \{4, 4, 4, 5, 6\}$, $E_3 = E_4 = \{2, 4\}$, $E_5 = \{2, 5, 6, 7\}$. Тогда $M(E_0; E_1, E_2, E_3, E_4, E_5)$ — сеть, геометрическая реализация которой изображена на следующем рисунке.



(П) Класс сетей с пустым E_0, каждый набор E_i $(i > 0)$ которых состоит из двух элементов, совпадает с классом графов. Таким образом, сети — это обобщение графов.

Вывод в классическом исчислении высказываний

(У) Правила введения логических связок:

1. (введение $\Rightarrow$) Из $\Gamma, A \vdash B$ следует $\Gamma \vdash A \Rightarrow B$ [теорема о дедукции];
2. (введение $\vee$) $A \vdash A \vee B$ (A8 и m.p.) и $B \vdash A \vee B$ [A9 и m.p.];
3. (введение $\&$) $A, B \vdash A \,\&\, B$ [A5 и m.p.];
4. (введение $\neg$) Из $\Gamma, A \vdash B$ и $\Gamma, A \vdash \neg B$ следует $\Gamma \vdash \neg A$ [
   1) $\Gamma, A \vdash B$;
   2) $\Gamma, A \vdash \neg B$;
   3) $\Gamma \vdash A \Rightarrow B$ (введение $\Rightarrow$ в 1);
   4) $\Gamma \vdash A \Rightarrow \neg B$ (введение $\Rightarrow$ в 2);
   5) $\vdash (A \Rightarrow B) \Rightarrow ((A \Rightarrow \neg B) \Rightarrow \neg A)$ (A3);
   6) $\Gamma \vdash (A \Rightarrow \neg B) \Rightarrow \neg A$ (m.p. к 3 и 5);
   7) $\Gamma \vdash \neg A$ (m.p. к 4 и 6)].

(У) Правила удаления логических связок:

1. (удаление $\Rightarrow$) $A, A \Rightarrow B \vdash B$ [m.p.];

2. (удаление &) $A \,\&\, B \vdash A$ и $A \,\&\, B \vdash B$ [А6, А7, удаление $\Rightarrow$];
3. (удаление $\neg$) $\neg\neg A \vdash A$ [А4 и 4-е свойство штопора] и $A, \neg A \vdash B$ [
   1) $A \vdash A$;
   2) $A, \neg A, \neg B \vdash A$ (1-свойство штопора);
   3) $\neg A \vdash \neg A$;
   4) $\neg A, A, \neg B \vdash \neg A$;
   5) $A, \neg A \vdash \neg\neg B$ (введение $\neg$ к 2 и 4);
   6) $A, \neg A \vdash B$];
4. (удаление $\vee$) Из $\Gamma \vdash A \vee B$, $\Gamma, A \vdash C$ и $\Gamma, B \vdash C$ следует $\Gamma \vdash C$ [
   1) $\Gamma \vdash A \vee B$;
   2) $\Gamma, A \vdash C$;
   3) $\Gamma, B \vdash C$;
   4) $\vdash (A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$ (А10);
   5) $\Gamma \vdash A \Rightarrow C$ (введение $\Rightarrow$ к 2);
   6) $\Gamma \vdash B \Rightarrow C$ (введение $\Rightarrow$ к 3);
   7) $\Gamma \vdash (B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C)$ (m.p. к 4 и 5);
   8) $\Gamma \vdash A \vee B \Rightarrow C$ (m.p. к 6 и 7);
   9) $\Gamma \vdash C$ (m.p. к 1 и 8)].

(П) Задача о Штирлице. Имеются две посылки: $\Phi_1 = (\text{МН} \,\&\, \text{АН} \vee \text{АН} \,\&\, \text{ШН}) \vee (\text{МН} \,\&\, \text{ШН})$; $\Phi_2 = (\text{МН} \Rightarrow \neg\text{ШН}) \,\&\, (\neg\text{ШН} \Rightarrow \text{МН})$. Докажем, что из Φ_1 и Φ_2 выводится АН, т.е. $\Phi_1, \Phi_2 \vdash \text{АН}$:

1) $\text{МН} \,\&\, \text{ШН} \vdash \text{МН}$ (удаление &);
2) $\text{МН} \,\&\, \text{ШН} \vdash \text{ШН}$ (удаление &);
3) $\Phi_2 \vdash \text{МН} \Rightarrow \neg\text{ШН}$ (удаление & к Φ_2);
4) $\Phi_2, \text{МН} \,\&\, \text{ШН} \vdash \neg\text{ШН}$ (m.p. к 1 и 3);
5) $\text{ШН}, \neg\text{ШН} \vdash \text{АН}$ (удаление $\neg$)
6) $\Phi_2, \text{МН} \,\&\, \text{ШН} \vdash \text{АН}$ (2-е свойство штопора к 2, 4, 5);
7) $\text{МН} \,\&\, \text{АН} \vdash \text{АН}$ (удаление &);
8) $\text{АН} \,\&\, \text{ШН} \vdash \text{АН}$ (удаление &);
9) $\vdash \text{МН} \,\&\, \text{АН} \Rightarrow \text{АН}$ (введение $\Rightarrow$ к 7);
10) $\Phi_2 \vdash \text{МН} \,\&\, \text{ШН} \Rightarrow \text{АН}$ (введение $\Rightarrow$ к 6);
11) $\vdash \text{АН} \,\&\, \text{ШН} \Rightarrow \text{АН}$ (введение $\Rightarrow$ к 8);
12) $\vdash (\text{МН} \,\&\, \text{АН} \Rightarrow \text{АН}) \Rightarrow ((\text{МН} \,\&\, \text{ШН} \Rightarrow \text{АН}) \Rightarrow (\text{МН} \,\&\, \text{АН} \vee \text{МН} \,\&\, \text{ШН} \Rightarrow \text{АН}))$ (А10);
13) $\vdash (\text{МН} \,\&\, \text{ШН} \Rightarrow \text{АН}) \Rightarrow (\text{МН} \,\&\, \text{АН} \vee \text{МН} \,\&\, \text{ШН} \Rightarrow \text{АН})$ (m.p. к 9 и 12);
14) $\Phi_2 \vdash \text{МН} \,\&\, \text{АН} \vee \text{МН} \,\&\, \text{ШН} \Rightarrow \text{АН}$ (m.p. к 10 и 13);
15) $\Phi_1, \Phi_2, \text{МН} \,\&\, \text{АН} \vee \text{АН} \,\&\, \text{ШН} \vdash \text{АН}$ (4-е и 1-е свойство штопора к 14);
16) $\Phi_1, \Phi_2 \vdash \Phi_1$;
17) $\Phi_1, \Phi_2 \vdash \text{АН}$ (удаление $\vee$ к 5, 14 и 15).

(П) Докажем закон исключения третьего $\vdash A \vee \neg A$:

1) $\neg(A \vee \neg A), A \vdash A \vee \neg A$ (введение $\vee$ и 1-е свойство штопора);
2) $\neg(A \vee \neg A), A \vdash \neg(A \vee \neg A)$ (1-е свойство штопора);
3) $\neg(A \vee \neg A) \vdash \neg A$ (введение $\neg$ к 1 и 2);
4) $\neg(A \vee \neg A), \neg A \vdash A \vee \neg A$ (введение $\vee$ и 1-е свойство штопора);
5) $\neg(A \vee \neg A), \neg A \vdash \neg(A \vee \neg A)$ (1-е свойство штопора);
6) $\neg(A \vee \neg A) \vdash \neg\neg A$ (введение $\neg$ к 4 и 5);
7) $\vdash \neg\neg(A \vee \neg A)$ (введение $\neg$ к 3 и 6);
8) $\vdash (A \vee \neg A)$ (удаление $\neg$).

Вывод в классическом исчислении предикатов 1-го порядка

(У) Правила введения кванторов:

1. (в.$\forall$) из $\Gamma \vdash A(x)$ следует $\Gamma \vdash \forall x A(x)$ — x не входит свободно в гипотезы из Γ [
   1) $\Gamma \vdash A(x)$;
   2) $\vdash A(x) \Rightarrow ((B \Rightarrow B) \Rightarrow A(x))$ (A1);
   3) $(B \Rightarrow B) \Rightarrow A(x)$ (m.p. 1,2);
   4) $\Gamma \vdash (B \Rightarrow B) \Rightarrow \forall x A(x)$ (R2);
   5) $\vdash B \Rightarrow B$ (известно);
   6) $\Gamma \vdash \forall x A(x)$ (m.p. 4,5)];
2. (в.$\exists$) $A(t) \vdash \exists x A(x)$ — t свободен для x в $A(x)$ [A12].

(У) Правила удаления кванторов:

1. (у.$\forall$) $\forall x A(x) \vdash A(t)$ — t свободен для x в $A(x)$ [A11];
2. (у.$\exists$) из $\Gamma, A(x) \vdash C$ следует $\Gamma, \exists x A(x) \vdash C$ — x не входит свободно в гипотезы из Γ и в C [
   1) $\Gamma, A(x) \vdash C$;
   2) $\Gamma \vdash A(x) \Rightarrow C$ (в.$\Rightarrow$);
   3) $\Gamma \vdash \exists x A(x) \Rightarrow C$ (R3);
   4) $\Gamma, \exists x A(x) \vdash C$ (4-е $\vdash$)].

(П) Докажем, что $\forall x \forall y A(x,y) \sim \forall y \forall x A(x,y)$ и $\exists x \exists y A(x,y) \sim \exists y \exists x A(x,y)$:

1) $A(x,y) \vdash \exists y \exists x A(x,y)$ (в.$\exists$ $\times 2$);
2) $\exists x \exists y A(x,y) \vdash \exists y \exists x A(x,y)$ (у.$\exists$ $\times 2$);
3) $\vdash \exists x \exists y A(x,y) \Rightarrow \exists y \exists x A(x,y)$ (в.$\Rightarrow$) и т.п.

Формулы $A(x)$ и $A(y)$ называются **подобными**, если $A(y)$ получается из $A(x)$ заменой всех свободных вхождений x на y, причем y свободен для x в $A(x)$ и не входит свободно в $A(x)$.

(У) Если $A(x)$ и $A(y)$ подобны, то $\forall x A(x) \sim \forall y A(y)$ и $\exists x A(x) \sim \exists y A(y)$. [Воспользуемся корректностью исчисления предикатов. Докажем 1-ю формулу ($\models \forall y A(y) \Rightarrow \forall x A(x)$ и $\models \forall x A(x) \Rightarrow \forall y A(y)$):

1) $\vdash \forall x A(x) \Rightarrow A(y)$ (A11);
2) $\vdash \forall x A(x) \Rightarrow \forall y A(y)$ (R2) и т.п.]

(П) Формулы $\forall x A(x,y)\ \&\ \exists y B(x,y)$ и $\forall v A(v,y)\ \&\ \exists w B(x,w)$ — эквивалентны.

(У) Если x не входит свободно ни в одну из гипотез из Γ, то из $\Gamma \vdash A(x) \sim B(x)$ следует $\Gamma \vdash \forall x A(x) \sim \forall x B(x)$ и $\Gamma \vdash \exists x A(x) \sim \exists x B(x)$ [

1) $\Gamma \vdash A(x) \Rightarrow B(x)$;
2) $\forall x A(x) \vdash A(x)$ (A11 и 4-е $\vdash$);
3) $\Gamma, \forall x A(x) \vdash B(x)$ (m.p. 1,2);
4) $\Gamma, \forall x A(x) \vdash \forall x B(x)$ (в.$\forall$);
5) $\Gamma \vdash \forall x A(x) \Rightarrow \forall x B(x)$ (в.$\Rightarrow$) и т.п.]

Верно и следующее более общее утверждение.

(У) **Теорема о замене.** Пусть $x_1, \ldots, x_n$ — переменные, свободно входящие в формулы A и B, которые становятся связанными одинаковым образом в формулах C_A и C_B. Тогда, если Γ множество гипотез, не содержащих свободных вхождений $x_1, \ldots, x_n$, то из $\Gamma \vdash A \sim B$ следует $\Gamma \vdash C_A \sim C_B$. []

Доказательства утверждений

(У) 1.[

1) $\neg\exists x A(x), A(x) \vdash \neg\exists x A(x)$
2) $\neg\exists x A(x), A(x) \vdash \exists x A(x)$ (в.$\exists$)
3) $\neg\exists x A(x) \vdash \neg A(x)$ (в.$\neg$)
4) $\neg\exists x A(x) \vdash \forall x \neg A(x)$ (в.$\forall$)
5) $\vdash \neg\exists x A(x) \Rightarrow \forall x \neg A(x)$ (в.$\Rightarrow$) — 1-я половина
6) $\forall x \neg A(x), A(x) \vdash \neg A(x)$ (у.$\forall$)
7) $\forall x \neg A(x), A(x) \vdash A(x)$
8) $A(x) \vdash \neg\forall x \neg A(x)$ (в.$\neg$)
9) $\exists x A(x) \vdash \neg\forall x \neg A(x)$ (у.$\exists$)
10) $\forall x \neg A(x), \exists x A(x) \vdash \forall x \neg A(x)$
11) $\forall x \neg A(x) \vdash \neg\exists x A(x)$ (у.$\neg$ 9,10)
12) $\vdash \forall x \neg A(x) \Rightarrow \neg\exists x A(x)$ (в.$\Rightarrow$) — 2-я половина];

2.[$\vdash \neg\forall x A(x) \sim \neg\forall x \neg\neg A(x) \sim \neg\neg\exists x \neg A(x) \sim \exists x \neg A(x)$];

3.[

1) $A\ \&\ \exists x B(x) \vdash A$ (у.&)
2) $A\ \&\ \exists x B(x) \vdash \exists x B(x)$ (у.&)
3) $A, B(x) \vdash A\ \&\ B(x)$ (в.&)
4) $A, B(x) \vdash \exists x (A\ \&\ B(x))$ (в.$\exists$)
5) $A\ \&\ \exists x B(x), B(x) \vdash \exists x (A\ \&\ B(x))$ (2-е $\vdash$ к 1)
6) $A\ \&\ \exists x B(x), \exists x B(x) \vdash \exists x (A\ \&\ B(x))$ (у.$\exists$)
7) $A\ \&\ \exists x B(x) \vdash \exists x (A\ \&\ B(x))$ (2-е $\vdash$ к 6)
8) $\vdash A\ \&\ \exists x B(x) \Rightarrow \exists x (A\ \&\ B(x))$ — 1-я половина
9) $A\ \&\ B(x) \vdash A$ (у.&)

10) $A \,\&\, B(x) \vdash B(x)$ (у.&)
11) $A \,\&\, B(x), B(x) \vdash \exists x B(x)$ (в.$\exists$)
12) $A \,\&\, B(x) \vdash B(x) \Rightarrow \exists x B(x)$ (в.$\Rightarrow$)
13) $A \,\&\, B(x) \vdash \exists x B(x)$ (m.p. 10,12)
14) $\vdash A \Rightarrow (\exists x B(x) \Rightarrow A \,\&\, \exists x B(x))$ (A5)
15) $A \,\&\, B(x) \vdash \exists x B(x) \Rightarrow A \,\&\, \exists x B(x)$ (m.p. 9,14)
16) $A \,\&\, B(x) \vdash A \,\&\, \exists x B(x)$ (m.p. 11,15)
17) $\exists x(A \,\&\, B(x)) \vdash A \,\&\, \exists x B(x)$ (у.$\exists$)
18) $\vdash \exists x(A \,\&\, B(x)) \Rightarrow A \,\&\, \exists x B(x)$ (в.$\Rightarrow$) — 2-я половина];

4.[
$\vdash A \vee \forall x B(x) \sim \neg(\neg A \,\&\, \neg\forall x B(x)) \sim \neg(\neg A \,\&\, \exists x \neg B(x)) \sim \neg\exists x(\neg A \,\&\, \neg B(x))\forall x\neg(\neg A \,\&\, \neg B(x)) \sim \forall x(A \vee B(x))$];

Машина Тьюринга

Конфигурацией машины Тьюринга называется совокупность, образованная: 1) словом на ленте; 2) внутренним состоянием; 3) позицией головки над лентой.