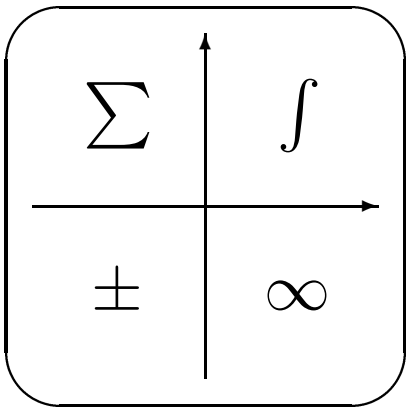
**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ**



А.В. Самохин

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

*Учебное пособие
для студентов II курса
специальности 220100*

Москва – 2003

МИНИСТЕРСТВО ТРАНСПОРТА РФ
ГОСУДАРСТВЕННАЯ СЛУЖБА
ГРАЖДАНСКОЙ АВИАЦИИ МОСКОВСКИЙ
ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ

Кафедра высшей математики

А.В. Самохин

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Учебное пособие
для студентов II курса
специальности 220100

Москва – 2003

Содержание

Предисловие

Имеет ли отношение математическая логика к тому, что необходимо знать специалисту по ЭВМ?

Мы надеемся, что в результате изучения этого курса слушатель убедится, что имеет, и самое непосредственное. Так, главы III и IV имеют отношение к алгоритмическому проектированию электронных схем; главы V и VI — к автоматическому порождению синтаксически правильных текстов, т.е. к специальному программированию; остаток книги посвящен основам теории алгоритмов: здесь обсуждаются, какие задачи вообще являются алгоритмически разрешимыми и какова сложность соответствующих алгоритмов. В главах I и II собран материал по началам теории множеств, необходимый для понимания остального текста (как, впрочем, и почти всей математики).

В основном тексте содержится более двухсот задач, в основном теоретической направленности, решение которых поможет разобраться в тонкостях теории; некоторые из них могут стать основой для научно-исследовательской работы студентов.

Задачи для практических занятий и контрольных заданий собраны в разделе "Задачи", составленном Ю. И. Дементьевым на основе [**4**]. Им же написаны программы на языке **C**.

В пособии использованы материалы из книг [**1**], [**2**] и [**3**] с любезного согласия авторов.

ГЛАВА I

Множества и мощности

§1. Множества

Основные понятия и обозначения, связанные с множествами и операциями над ними:

- *Множества* состоят из *элементов*. Запись $x \in M$ означает, что x является элементом множества M.
- Говорят, что множество A является *подмножеством* множества B (запись: $A \subset B$), если все элементы A являются элементами B.
- Множества A и B *равны* (запись: $A = B$), если они содержат одни и те же элементы (другими словами, если $A \subset B$ и $B \subset A$).
- Если A — подмножество B, не равное всему B, то A называют *собственным* подмножеством B (запись: $A \subsetneq B$).
- *Пустое* множество $\varnothing$ не содержит ни одного элемента и является подмножеством любого множества.
- *Пересечение* $A \cap B$ двух множеств A и B состоит из элементов, которые принадлежат обоим множествам A и B. Это записывают так:

  $$A \cap B = \{x \mid x \in A \text{ и } x \in B\}$$

  (читается: множество таких x, что $\dots$).
- *Объединение* $A \cup B$ состоит из элементов, которые принадлежат хотя бы одному из множеств A и B:

  $$A \cup B = \{x \mid x \in A \text{ или } x \in B\}.$$

- *Разность* $A \setminus B$ состоит из элементов, которые принадлежат A, но не принадлежат B:

  $$A \setminus B = \{x \mid x \in A \text{ и } x \notin B\}.$$

  Если множество B является подмножеством множества A, разность $A \setminus B$ называют также *дополнением B до A*.
- *Симметрическая разность* $A \triangle B$ состоит из элементов, которые принадлежат ровно одному из множеств A и B:

  $$A \triangle B = (A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B).$$

- Через $\{a, b, c\}$ обозначается множество, которое содержит элементы a, b, c и не содержит других. Если среди a, b, c есть равные, оно

может содержать один или два элемента. Подобное обозначение используется и в менее формальных ситуациях: множество членов последовательности $a_0, a_1, \ldots$ обозначается $\{a_0, a_1, \ldots\}$ или даже $\{a_i\}$. Более аккуратная запись для того же множества такова: $\{a_i \mid i \in \mathbb{N}\}$, где $\mathbb{N}$ — множество натуральных чисел $\{0, 1, 2, \ldots\}$.

Понятие множества появилось в математике сравнительно недавно, в конце 19-го века, в связи с работами Кантора (сравнение мощностей множеств), о которых пойдёт речь дальше (раздел 3 и следующие). Некоторое время назад этот язык пытались внедрить в школьное преподавание, объясняя ученикам, что у уравнения $x^2 + 1 = 0$ есть множество решений (впрочем, пустое), что множество решений системы уравнений есть пересечение множеств решений каждого из них (а для совокупности уравнений — объединение), что в множестве $\{2, 2, 3\}$ не три элемента, а два, и оно равно множеству $\{2, 3\}$, что $\varnothing$, $\{\varnothing\}$ и $\{\varnothing, \{\varnothing\}\}$ — это три совершенно разных множества и т. д. Но всё равно большинство школьников так и не поняло, почему множество решений уравнения $x^2 = 4$ можно записывать как $\{-2, 2\}$, а множество решений уравнения $x^2 = -4$ нельзя записывать как $\{\varnothing\}$ (а надо писать $\varnothing$). Отметим кстати ещё два расхождения: в школе натуральные числа начинаются с единицы, а в некоторых книжках — с нуля (мы тоже будем называть нуль натуральным числом). Кроме того, иногда вместо $\subset$ пишут $\subseteq$, используя $\subset$ для собственных подмножеств (вместо нашего $\subsetneq$).

Мы предполагаем, что перечисленные выше основные понятия теории множеств более или менее вам знакомы, и будем достаточно свободно ими пользоваться. Вот несколько задач для самоконтроля; надеемся, что большинство из них не представит для вас большого труда.

Задача 1. *Старейший математик среди шахматистов и старейший шахматист среди математиков — это один или тот же человек или (возможно) разные?*

Задача 2. *Лучший математик среди шахматистов и лучший шахматист среди математиков — это один или тот же человек или (возможно) разные?*

Задача 3. *Каждый десятый математик — шахматист, а каждый шестой шахматист — математик. Кого больше — математиков или шахматистов — и во сколько раз?*

Задача 4. *Существуют ли такие множества A, B и C, что $A \cap B \neq \varnothing$, $A \cap C = \varnothing$ и $(A \cap B) \setminus C = \varnothing$?*

Задача 5. *Какие из равенств (***а***)* $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$*; (***б***)* $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$*; (***в***)* $(A \cup B) \setminus C = (A \setminus C) \cup B$*; (***г***)* $(A \cap B) \setminus C = (A \setminus C) \cap B$*; (***д***)* $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$*; (***е***)* $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$ *верны для любых множеств* A, B, C*?*

Задача 6. *Проведите подробное доказательство верных равенств предыдущей задачи, исходя из определений. (Докажем, что множества в левой и правой частях равны. Пусть* x *— любой элемент левой части равенства. Тогда... Поэтому* x *входит в правую часть. Обратно, пусть...) Приведите контрпримеры к неверным равенствам.*

Задача 7. *Докажите, что симметрическая разность ассоциативна:* $A \triangle (B \triangle C) = (A \triangle B) \triangle C$ *для любых* A*,* B *и* C*. (Указание: сложение по модулю 2 ассоциативно.)*

Задача 8. *Сколько различных выражений для множеств можно составить из переменных* A *и* B *с помощью (многократно используемых) операций пересечения, объединения и разности? (Два выражения считаются одинаковыми, если они равны при любых значениях переменных.) Тот же вопрос для трёх множеств и для* n *множеств. (Ответ в общем случае:* 2^{2^n-1}*.)*

Задача 9. *Тот же вопрос, если используются только операции* $\cup$ *и* $\cap$*. (Для двух и трёх переменных это число несложно подсчитать, но общей формулы для* n *переменных не известно. Эту задачу называют также задачей о числе монотонных булевых функций от* n *аргументов.)*

Задача 10. *Сколько существует подмножеств у* n*-элементного множества?*

Задача 11. *Пусть множество* A *содержит* n *элементов, а его подмножество* B *содержит* k *элементов. Сколько существует множеств* C*, для которых* $B \subset C \subset A$*?*

Задача 12. *Множество* U *содержит* $2n$ *элементов. В нём выделено* k *подмножеств, причём ни одно из них не является подмножеством другого. Каково может быть максимальное значение числа* k*?*

§2. Число элементов

Число элементов в конечном множестве A называют также его *мощностью* и обозначают $|A|$ (а также $\#A$). (Вскоре мы будем говорить о мощностях и для бесконечных множеств.) Следующая формула позволяет найти

мощность объединения нескольких множеств, если известны мощности каждого из них, а также мощности всех пересечений.

Теорема 1 (Формула включений и исключений).

$$\begin{aligned} |A \cup B| &= |A| + |B| - |A \cap B|; \\ |A \cup B \cup C| &= |A| + |B| + |C| - \\ &\quad - |A \cap B| - |A \cap C| - |B \cap C| + \\ &\quad + |A \cap B \cap C|; \end{aligned}$$

вообще $|A_1 \cup \ldots \cup A_n|$ *равно*

$$\sum_i |A_i| - \sum_{i<j} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k| - \ldots$$

Доказательство. Это утверждение несложно доказать индукцией по n, но мы приведём другое доказательство. Фиксируем произвольное множество U, подмножествами которого являются множества $A_1, \ldots, A_n$.

Характеристической функцией множества $X \subset U$ называют функцию χ_X, которая равна 1 на элементах X и 0 на остальных элементах U. Операции над подмножествами множества U соответствуют операциям с их характеристическими функциями. В частности, пересечению множеств соответствует произведение характеристических функций:

$$\chi_{A \cap B}(u) = \chi_A(u)\chi_B(u)$$

Дополнению (до U) соответствует функция $1 - \chi$, если χ — характеристическая функция исходного множества.

Число элементов множества можно записать как сумму значений его характеристической функции:

$$|X| = \sum_u \chi_X(u).$$

Объединение $A_1 \cup \ldots \cup A_N$ можно записать как дополнение к пересечению дополнений множеств A_i; в терминах характеристических функций имеем

$$\chi_{A_1 \cup \ldots \cup A_n} = 1 - (1 - \chi_{A_1}) \ldots (1 - \chi_{A_n}).$$

Раскрыв скобки в правой части, мы получим

$$\sum_i \chi_{A_i} - \sum_{i<j} \chi_{A_i}\chi_{A_j} + \sum_{i<j<k} \chi_{A_i}\chi_{A_j}\chi_{A_k} - \ldots$$

и просуммировав левую и правую часть по всем элементам U (обе они есть функции на U), получим формулу включений и исключений. □

Задача 13. *Докажите, что* $|A_1 \triangle \ldots \triangle A_n|$ *равно*

$$\sum_i |A_i| - 2\sum_{i<j} |A_i \cap A_j| + 4 \sum_{i<j<k} |A_i \cap A_j \cap A_k| - \ldots$$

(коэффициенты — последовательные степени двойки).

Подсчёт количеств элементов в конечных множествах относят к *комбинаторике*. Некоторые начальные сведения из комбинаторики приведены дальше в качестве задач. Сейчас нас в первую очередь интересует следующий принцип:

если между двумя множествами можно установить взаимно однозначное соответствие, то в них одинаковое число элементов.

(Взаимная однозначность требует, чтобы каждому элементу первого множества соответствовал ровно один элемент второго и наоборот.)

Вот несколько примеров использования этого принципа.

Задача 14. *На окружности выбраны 1000 белых точек и одна чёрная. Чего больше — треугольников с вершинами в белых точках или четырёхугольников, у которых одна вершина чёрная, а остальные три белые? (Решение: их поровну, поскольку каждому четырёхугольнику соответствует треугольник, образованный тремя его белыми вершинами.)*

Задача 15. *Каких подмножеств больше у* 100*-элементного множества: мощности* 57 *или мощности* 43*? (Указание:* $57 + 43 = 100$*.)*

Задача 16. *Докажите, что последовательностей длины* n*, составленных из нулей и единиц, столько же, сколько подмножеств у множества* $\{1, 2, \ldots, n\}$*. (Указание: каждому подмножеству* $X \subset \{1, 2, \ldots, n\}$ *соответствует характеристическая последовательность, на* i*-м месте которой стоит единица, если и только если* $i \in X$*.)*

Задача 17. *Докажите, что последовательностей нулей и единиц длины* n*, в которых число единиц равно* k*, равно числу* k*-элементных подмножеств* n*-элементного множества.*

Это число называется *числом сочетаний из* n *по* k и обозначается C_n^k в русских книжках; в иностранных обычно используется обозначение $\binom{n}{k}$.

Задача 18. *Докажите, что* $C_n^k = C_n^{n-k}$.

Задача 19. *Докажите, что* $C_n^0 + C_n^1 + \ldots + C_n^n = 2^n$.

Задача 20. *Пусть* U *— непустое конечное множество. Докажите, что подмножеств множества* U*, имеющих чётную мощность, столько*

же, сколько имеющих нечётную мощность. (Указание: фиксируем элемент $u \in U$ и объединим в пары подмножества, отличающиеся только в точке u.)

Задача 21. *Докажите, что* $C_n^0 - C_n^1 + C_n^2 - \ldots + (-1)^n C_n^n = 0$. *(Указание: как это связано с предыдущей задачей?)*

Задача 22. *Докажите формулу* бинома Ньютона*:*

$$(a+b)^n = C_n^0 a^n + C_n^1 a^{n-1} b + \ldots + C_n^k a^{n-k} b^k + \ldots + C_n^n b^n.$$

Задача 23. *Докажите, что способов расстановки скобок (указывающих порядок действий) в неассоциативном произведении из n элементов столько же, сколько способов разбить выпуклый $(n+1)$-угольник на треугольники непересекающимися диагоналями. (Для произведения трёх множителей есть два варианта $(ab)c$ и $a(bc)$; с другой стороны, есть два способа разрезать четырёхугольник на два треугольника, проведя диагональ. Для произведения четырёх сомножителей и для пятиугольника имеется по 5 вариантов.)*

§3. Равномощные множества

Два множества называют *равномощными*, если между ними можно установить взаимно однозначное соответствие, при котором каждому элементу одного множества соответствует ровно один элемент другого.

Для конечных множеств это означает, что в них одинаковое число элементов, но определение имеет смысл и для бесконечных множеств. Например, отрезки $[0,1]$ и $[0,2]$ равномощны, поскольку отображение $x \mapsto 2x$ осуществляет искомое соответствие.

Задача 24. *Докажите, что любые два интервала (a,b) и (c,d) на прямой равномощны.*

Задача 25. *Докажите, что любые две окружности на плоскости равномощны. Докажите, что любые два круга на плоскости равномощны.*

Задача 26. *Докажите, что полуинтервал $[0,1)$ равномощен полуинтервалу $(0,1]$.*

Несколько более сложна такая задача: доказать, что интервал $(0,1)$ и луч $(0,+\infty)$ равномощны. Это делается так. Заметим, что отображение $x \mapsto 1/x$ является взаимно однозначным соответствием между $(0,1)$ и $(1,+\infty)$, а $x \mapsto$ $\mapsto (x-1)$ — взаимно однозначным соответствием между $(1,+\infty)$ и $(0,+\infty)$,

поэтому их композиция $x \mapsto (1/x) - 1$ является искомым взаимно однозначным соответствием между $(0, 1)$ и $(0, +\infty)$.

Вообще, как говорят, отношение равномощности есть *отношение эквивалентности*. Это означает, что оно *рефлексивно* (каждое множество равномощно самому себе), *симметрично* (если A равномощно B, то и B равномощно A) и *транзитивно* (если A равномощно B и B равномощно C, то A равномощно C). Свойством транзитивности мы только что воспользовались, взяв луч $(1, +\infty)$ в качестве промежуточного множества.

Ещё несколько примеров:

- Множество бесконечных последовательностей нулей и единиц равномощно множеству всех подмножеств натурального ряда. (В самом деле, сопоставим с каждой последовательностью множество номеров мест, на которых стоят единицы: например, последовательность из одних нулей соответствует пустому множеству, из одних единиц — натуральному ряду, а последовательность 10101010 ... — множеству чётных чисел.)
- Множество бесконечных последовательностей цифр 0, 1, 2, 3 равномощно множеству бесконечных последовательностей цифр 0 и 1. (В самом деле, можно закодировать цифры 0, 1, 2, 3 группами 00, 01, 10, 11. Обратное преобразование разбивает последовательность нулей и единиц на пары, после чего каждая пара заменяется на цифру от 0 до 3.)
- Множество бесконечных последовательностей цифр 0, 1, 2 равномощно множеству бесконечных последовательностей цифр 0 и 1. (Можно было бы пытаться рассуждать так: это множество заключено между двумя множествами одной и той же мощности, и потому равномощно каждому из них. Этот ход мыслей правилен, как показывает теорема Кантора – Бернштейна из раздела 5. Но здесь можно обойтись и без этой теоремы, если закодировать цифры 0, 1 и 2 последовательностями 0, 10 и 11: легко сообразить, что всякая последовательность нулей и единиц однозначно разбивается на такие блоки слева направо. Такой способ кодирования называют префиксным кодом.)
- Пример с последовательностями нулей и единиц можно обобщить: множество подмножеств любого множества U (оно обычно обозначается $P(U)$ и по-английски называется power set) равномощно множеству всех функций, которые ставят в соответствие каждому элементу $x \in U$ одно из чисел 0 и 1 (множество таких функций обычно обозначают 2^X). (В самом деле, каждому множеству $X \subset U$ соответствует его характеристическая функция.)

Мы продолжим этот список, но сначала полезно доказать несколько простых фактов о счётных множествах (равномощных множеству натуральных чисел).

§4. Счётные множества

Множество называется *счётным*, если оно равномощно множеству $\mathbb{N}$ натуральных чисел, то есть если его можно представить в виде $\{x_0, x_1, x_2, \dots\}$ (здесь x_i — элемент, соответствующий числу i; соответствие взаимно однозначно, так что все x_i различны).

Например, множество целых чисел $\mathbb{Z}$ счётно, так как целые числа можно расположить в последовательность $0, 1, -1, 2, -2, 3, -3, \dots$

Теорема 2. *(а) Подмножество счётного множества конечно или счётно.*

(б) Всякое бесконечное множество содержит счётное подмножество.

(в) Объединение конечного или счётного числа конечных или счётных множеств конечно или счётно.

Доказательство. (а) Пусть B — подмножество счётного множества $A = \{a_0, a_1, a_2, \dots\}$. Выбросим из последовательности $a_0, a_1, \dots$ те члены, которые не принадлежат B (сохраняя порядок оставшихся). Тогда оставшиеся члены образуют либо конечную последовательность (и тогда B конечно), либо бесконечную (и тогда B счётно).

(б) Пусть A бесконечно. Тогда оно непусто и содержит некоторый элемент b_0. Будучи бесконечным, множество A не исчерпывается элементом b_0 — возьмём какой-нибудь другой элемент b_1, и т. д. Получится последовательность $b_0, b_1, \dots$; построение не прервётся ни на каком шаге, поскольку A бесконечно. Теперь множество $B = \{b_0, b_1, \dots\}$ и будет искомым счётным подмножеством. (Заметим, что B вовсе не обязано совпадать с A, даже если A счётно.)

(в) Пусть имеется счётное число счётных множеств $A_1, A_2, \dots$ Расположив элементы каждого из них слева направо в последовательность ($A_i = \{a_{i0}, a_{i1}, \dots\}$) и поместив эти последовательности друг под другом, получим таблицу

$$\begin{array}{ccccc} a_{00} & a_{01} & a_{02} & a_{03} & \dots \\ a_{10} & a_{11} & a_{12} & a_{13} & \dots \\ a_{20} & a_{21} & a_{22} & a_{23} & \dots \\ a_{30} & a_{31} & a_{32} & a_{33} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{array}$$

Теперь эту таблицу можно развернуть в последовательность, например, проходя по очереди диагонали:

$$a_{00},\ a_{01}, a_{10},\ a_{02}, a_{11}, a_{20},\ a_{03}, a_{12}, a_{21}, a_{30}, \ldots$$

Если множества A_i не пересекались, то мы получили искомое представление для их объединения. Если пересекались, то из построенной последовательности надо выбросить повторения.

Если множеств конечное число или какие-то из множеств конечны, то в этой конструкции части членов не будет — и останется либо конечное, либо счётное множество. □

Задача 27. *Описанный проход по диагоналям задаёт взаимно однозначное соответствие между множеством всех пар натуральных чисел (которое обозначается* $\mathbb{N} \times \mathbb{N}$*) и* $\mathbb{N}$*. Любопытно, что это соответствие задаётся простой формулой (многочленом второй степени с рациональными коэффициентами). Укажите этот многочлен.*

Замечание. В доказательстве утверждения (б) теоремы 2 есть тонкий момент: на каждом шаге мы должны выбрать один из оставшихся элементов множества A; такие элементы есть, но у нас нет никакого правила, позволяющего такой выбор описать. При более формальном построении теории множеств тут нужно сослаться на специальную аксиому, называемую *аксиомой выбора*. Законность этой аксиомы вызывала большие споры в начале 20-го века, но постепенно к ней привыкли, и эти споры сейчас почти не воспринимаются. В середине века великий логик Курт Гёдель доказал, что аксиому выбора нельзя опровергнуть, пользуясь остальными аксиомами теории множеств, а в 1960-е годы американский математик Пол Дж. Коэн доказал, что её нельзя и вывести из остальных аксиом. (Конечно, понимание этих утверждений требует подробного изложения теории множеств как аксиоматической теории.)

Ещё несколько примеров счётных множеств:

- Множество $\mathbb{Q}$ рациональных чисел счётно. В самом деле, рациональные числа представляются несократимыми дробями с целым числителем и знаменателем. Множество дробей с данным знаменателем счётно, поэтому $\mathbb{Q}$ представимо в виде объединения счётного числа счётных множеств. Забегая вперёд (см. раздел 6), отметим, что множество $\mathbb{R}$ всех действительных чисел несчётно.
- Множество $\mathbb{N}^k$, элементами которого являются наборы из k натуральных чисел, счётно. Это легко доказать индукцией по k. При $k = 2$ множество $\mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$ пар натуральных чисел разбивается на счётное число счётных множеств $\{0\} \times \mathbb{N}, \{1\} \times \mathbb{N}, \ldots$ (элементами

i-го множества будут пары, первый член которых равен i). Поэтому $\mathbb{N}^2$ счётно. Аналогичным образом множество $\mathbb{N}^3$ троек натуральных чисел разбивается на счётное число множеств $\{i\} \times \mathbb{N} \times \mathbb{N}$. Каждое из них состоит из троек, первый член которых фиксирован и потому равномощно множеству $\mathbb{N}^2$, которое счётно. Точно так же можно перейти от счётности множества $\mathbb{N}^k$ к счётности множества $\mathbb{N}^{k+1}$.

- Множество всех конечных последовательностей натуральных чисел счётно. В самом деле, множество всех последовательностей данной длины счётно (как мы только что видели), так что интересующее нас множество разбивается на счётное число счётных множеств.
- В предыдущем примере не обязательно говорить о натуральных числах — можно взять любое счётное (или конечное) множество. Например, множество всех текстов, использующих русский алфавит (такой текст можно считать конечной последовательностью букв, пробелов, знаков препинания и т. п.), счётно; то же самое можно сказать о множестве (всех мыслимых) компьютерных программ и т. д.
- Число называют *алгебраическим*, если оно является корнем ненулевого многочлена с целыми коэффициентами. Множество алгебраических чисел счётно, так как многочленов счётное число (многочлен задаётся конечной последовательностью целых чисел — его коэффициентов), а каждый многочлен имеет конечное число корней (не более n для многочленов степени n).
- Множество периодических дробей счётно. В самом деле, такая дробь может быть записана как конечная последовательность символов из конечного множества (запятая, цифры, скобки); например, дробь 0,16666... можно записать как 0,1(6). А таких последовательностей счётное множество.

Задача 28. *Докажите, что любое семейство непересекающихся интервалов на прямой конечно или счётно. (Указание: в каждом интервале найдётся рациональная точка.)*

Задача 29. *(**а**) Докажите, что любое множество непересекающихся восьмёрок на плоскости конечно или счётно. (Восьмёрка — объединение двух касающихся окружностей любых размеров.) (**б**) Сформулируйте и докажите аналогичное утверждение для букв* **Т**.

Задача 30. *Докажите, что множество точек строгого локального максимума любой функции действительного аргумента конечно или счётно.*

Задача 31. *Докажите, что множество точек разрыва неубывающей*

функции действительного аргумента конечно или счётно.

Теорема 3. *Если множество A бесконечно, а множество B конечно или счётно, то объединение $A \cup B$ равномощно A.*

Доказательство. Можно считать, что B не пересекается с A (пересечение можно выбросить из B, останется по-прежнему конечное или счётное множество).

Выделим в A счётное подмножество P; остаток обозначим через Q. Тогда нам надо доказать, что $B+P+Q$ равномощно $P+Q$ (знак $+$ символизирует объединение непересекающихся множеств). Поскольку $B+P$ и P оба счётны, между ними существует взаимно однозначное соответствие. Его легко продолжить до соответствия между $B+P+Q$ и $P+Q$ (каждый элемент множества Q соответствует сам себе). □

Задача 32. *Примените эту конструкцию и явно укажите соответствие между отрезком $[0,1]$ и полуинтервалом $[0,1)$.*

Задача 33. *Теорема 3 показывает, что добавление счётного множества к бесконечному не меняет его мощности. Можно ли сказать то же самое про удаление? Докажите, что если A бесконечно и не является счётным, а B конечно или счётно, то $A \setminus B$ равномощно B.*

Задача 34. *Немецкий математик Р. Дедекинд предложил такое определение бесконечного множества: множество бесконечно, если оно равномощно некоторому своему подмножеству, не совпадающему со всем множеством. Покажите, что указанное Дедекиндом свойство действительно определяет бесконечные множества.*

Добавляя конечные или счётные множества, легко понять, что прямая, все промежутки на прямой (отрезки, интервалы, полуинтервалы), лучи, их конечные или счётные объединения и т. п. равномощны друг другу.

Задача 35. *Укажите взаимно однозначное соответствие между множеством $[0,1] \cup [2,3] \cup [4,5] \cup \ldots$ и отрезком $[0,1]$.*

Задача 36. *Докажите, что множество всех прямых на плоскости равномощно множеству всех точек на плоскости. (Указание: и точки, и прямые задаются парами чисел — за небольшими исключениями.)*

Задача 37. *Докажите, что полуплоскость (точки плоскости, лежащие по одну сторону от некоторой прямой) равномощна плоскости. (Это верно независимо от того, включаем мы граничную прямую в полуплоскость или нет.)*

Теорема 4. *Отрезок* $[0,1]$ *равномощен множеству всех бесконечных последовательностей нулей и единиц.*

Доказательство. В самом деле, каждое число $x \in [0,1]$ записывается в виде бесконечной двоичной дроби. Первый знак этой дроби равен 0 или 1 в зависимости от того, попадает ли число x в левую или правую половину отрезка. Чтобы определить следующий знак, надо выбранную половину поделить снова пополам и посмотреть, куда попадёт x, и т. д.

Это же соответствие можно описать в другую сторону: последовательности $x_0x_1x_2\ldots$ соответствует число, являющееся суммой ряда

$$\frac{x_0}{2}+\frac{x_1}{4}+\frac{x_2}{8}+\ldots$$

(В этом построении мы используем некоторые факты из математического анализа, что не удивительно — нас интересуют свойства действительных чисел.)

Описанное соответствие пока что не совсем взаимно однозначно: двоично-рациональные числа (дроби вида $m/2^n$) имеют два представления. Например, число $3/8$ можно записать как в виде 0,011000..., так и в виде 0,010111... Соответствие станет взаимно однозначным, если отбросить дроби с единицей в периоде. Но таких дробей счётное число, поэтому на мощность это не повлияет. □

Задача 38. *Какая двоичная дробь соответствует числу* $1/3$*?*

В этом доказательстве можно было бы использовать более привычные десятичные дроби вместо двоичных. Получилось бы, что отрезок $[0,1]$ равномощен множеству всех бесконечных последовательностей цифр $0,1,\ldots,9$. Чтобы перейти отсюда к последовательностям нулей и единиц, можно воспользоваться приёмом, описанным на с. 13.

Теперь всё готово для доказательства такого удивительного факта:

Теорема 5. *Квадрат (со внутренностью) равномощен отрезку.*

Доказательство. Квадрат равномощен множеству $[0,1]\times[0,1]$ пар действительных чисел, каждое из которых лежит на отрезке $[0,1]$ (метод координат). Мы уже знаем, что вместо чисел на отрезке можно говорить о последовательностях нулей и единиц. Осталось заметить, что паре последовательностей нулей и единиц $\langle x_0x_1x_2\ldots, y_0y_1y_2\ldots\rangle$ можно поставить в соответствие последовательность-смесь $x_0y_0x_1y_1x_2y_2\ldots$ и что это соответствие будет взаимно однозначным. □

Этот результат был получен в 1877 году немецким математиком Георгом Ка́нтором и удивил его самого, поскольку противоречил интуитивному ощущению размерности (квадрат двумерен, поэтому вроде бы должен содержать больше точек, чем одномерный отрезок). Результат Кантора не лишает смысла понятие размерности, если рассматривать лишь непрерывные в обе стороны соответствия, и тогда пространства разной размерности можно будет различить. (Заметим также, что существует непрерывное отображение отрезка в квадрат, которое проходит через любую точку квадрата. Оно называется кривой Пеано.)

Из теоремы 5 легко получить много других утверждений про равномощность геометрических объектов: круг равномощен окружности, прямая равномощна плоскости и т. п.

Можно также заметить, что пространство (точки которого задаются тремя координатами $\langle x, y, z\rangle$) равномощно плоскости (надо закодировать пару $\langle x, y\rangle$ одним числом), и, следовательно, прямой. То же самое можно проделать и для пространств большей размерности.

Задача 39. *Докажите, что множество всех конечных последовательностей действительных чисел равномощно* $\mathbb{R}$ *(множеству всех действительных чисел).*

Задача 40. *Докажите, что множество всех бесконечных последовательностей действительных чисел равномощно* $\mathbb{R}$.

Отметим, что мы пока не умеем доказывать, что множество действительных чисел (или множество бесконечных последовательностей нулей и единиц) несчётно. Это будет сделано в разделе 6.

Мощность множества действительных чисел называют *мощностью континуума* (от латинского слова, означающего непрерывный; имеется в виду, что точка на отрезке может непрерывно двигаться от одного конца к другому).

§5. Теорема Кантора – Бернштейна

Определение равномощности уточняет интуитивную идею о множествах одинакового размера. А как формально определить, когда одно множество больше другого?

Говорят, что множество A *по мощности не больше* множества B, если оно равномощно некоторому подмножеству множества B (возможно, самому B).

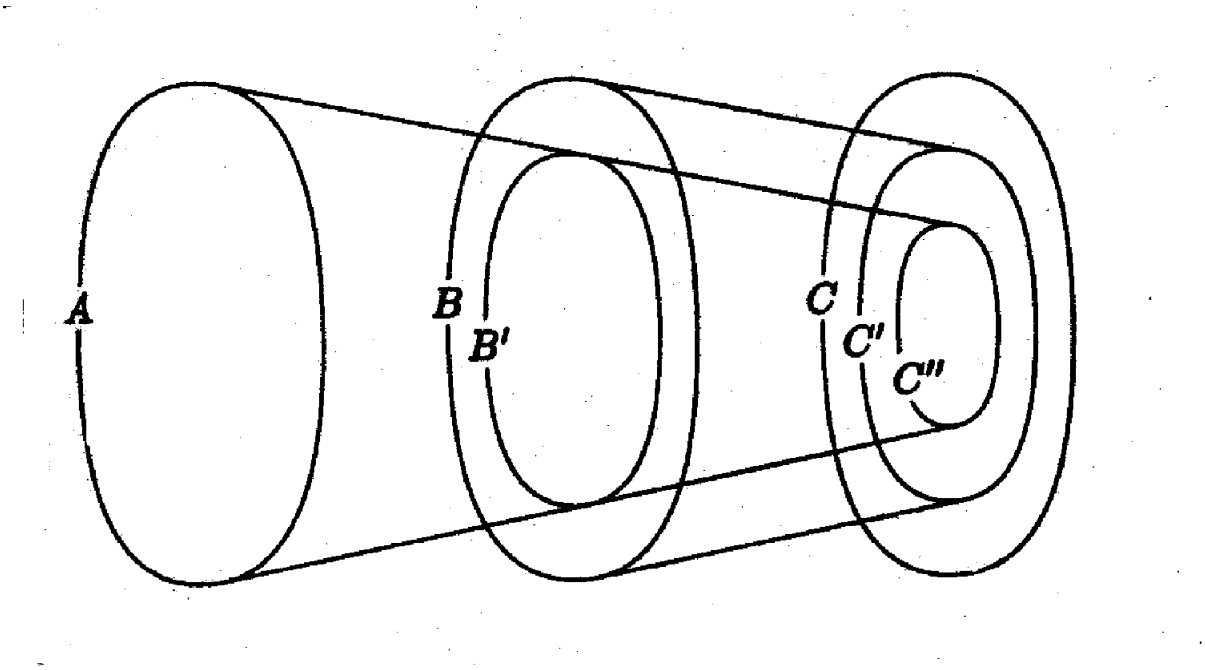


Рис. 1. Транзитивность сравнения мощностей

Задача 41. *Некто предложил такое определение: множество A имеет строго меньшую мощность, чем множество B, если оно равномощно некоторой части множества B, не совпадающей со всем B. Почему это определение неудачно? (Указание. Популярные рассказы о теории множеств часто начинаются с такого парадокса, восходящего к Галилею. Каких чисел больше — всех натуральных чисел или точных квадратов? С одной стороны, точные квадраты составляют лишь небольшую часть натуральных чисел; с другой стороны их можно поставить во взаимно однозначное соответствие со всеми натуральными числами.)*

Отношение иметь не большую мощность обладает многими естественными свойствами:

- Если A и B равномощны, то A имеет не большую мощность, чем B. (Очевидно.)
- Если A имеет не большую мощность, чем B, а B имеет не большую мощность, чем C, то A имеет не большую мощность, чем C. (Тоже несложно. Пусть A находится во взаимно однозначном соответствии с $B' \subset B$, а B находится во взаимно однозначном соответствии с $C' \subset \subset C$. Тогда при втором соответствии B' соответствует некоторому множеству $C'' \subset C' \subset C$, как показано на рис. 5, и потому A равномощно C''.)
- Если A имеет не большую мощность, чем B, а B имеет не большую мощность, чем A, то они равномощны. (Это вовсе не очевидное

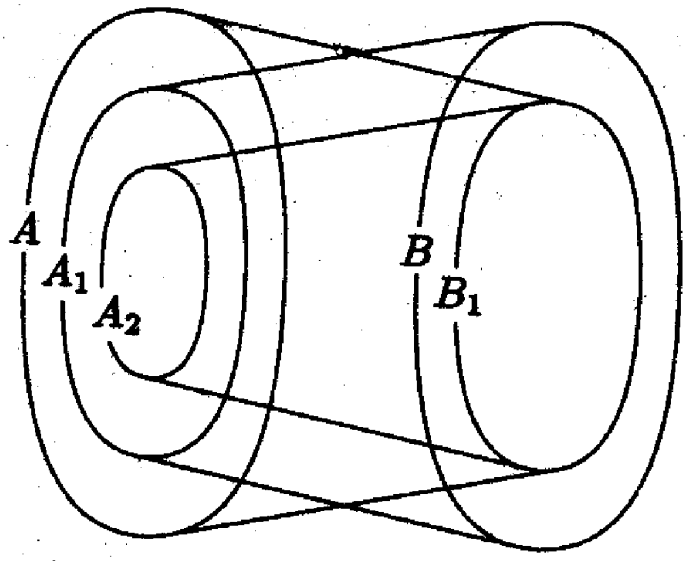


Рис. 2

утверждение составляет содержание теоремы Кантора – Бернштейна, которую мы сейчас докажем.)

- Для любых двух множеств A и B верно (хотя бы) одно из двух: либо A имеет не большую мощность, чем B, либо B имеет не большую мощность, чем A. (Доказательство этого факта требует так называемой трансфинитнойиндукции и здесь принимается на веру)

Теорема 6 (Кантора – Бернштейна). *Если множество A равномощно некоторому подмножеству множества B, а B равномощно некоторому подмножеству множества A, то множества A и B равномощны.*

Доказательство. Пусть A равномощно подмножеству B_1 множества B, а B равномощно подмножеству A_1 множества A (см. рис. 2). При взаимно однозначном соответствии между B и A_1 подмножество $B_1 \subset B$ переходит в некоторое подмножество $A_2 \subset A_1$. При этом все три множества A, B_1 и A_2 равномощны, — и нужно доказать, что они равномощны множеству B, или, что то же самое, A_1.

Теперь мы можем забыть про множество B и его подмножества и доказывать такой факт: *если $A_2 \subset A_1 \subset A_0$ и A_2 равномощно A_0, то все три множества равномощны.* (Для единообразия мы говорим A_0 вместо A.)

Пусть f — функция, осуществляющая взаимно однозначное соответствие $A_0 \to A_2$ (элемент $x \in A_0$ соответствует элементу $f(x) \in A_2$). Когда A_0 переходит в A_2, меньшее множество A_1 переходит в какое-то множество $A_3 \subset \subset A_2$ (см. рис. 3). Аналогичным образом само A_2 переходит в некоторое

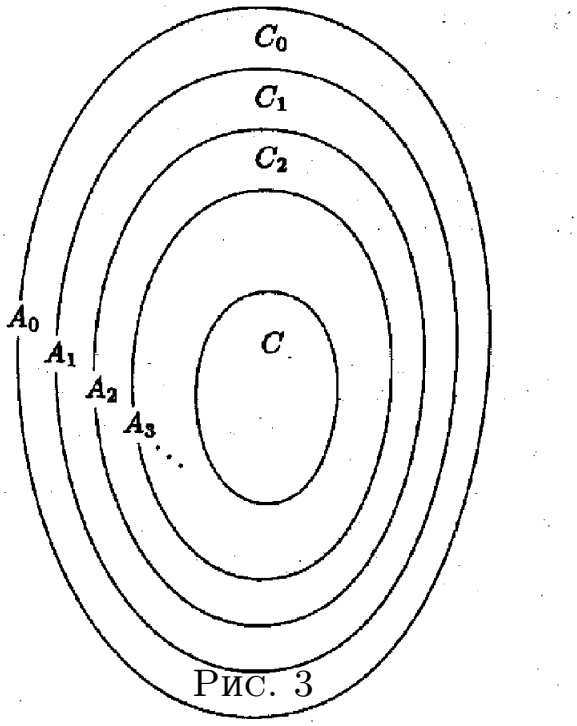


Рис. 3

множество $A_4 \subset A_2$. При этом $A_4 \subset A_3$, так как $A_1 \subset A_2$.

Продолжая эту конструкцию, мы получаем убывающую последовательность множеств

$$A_0 \supset A_1 \supset A_2 \supset A_3 \supset A_4 \supset \dots$$

и взаимно однозначное соответствие $f\colon A_0 \to A_2$, при котором A_i соответствует A_{i+2} (иногда это записывают так: $f(A_i) = A_{i+2}$). Формально можно описать A_{2n} как множество тех элементов, которые получаются из какого-то элемента множества A_0 после n-кратного применения функции f. Аналогичным образом A_{2n+1} состоит из тех и только тех элементов, которые получаются из какого-то элемента множества A_1 после n-кратного применения функции f.

Заметим, что пересечение всех множеств A_i вполне может быть непусто: оно состоит из тех элементов, у которых можно сколько угодно раз брать f-прообраз. Теперь можно сказать так: множество A_0 мы разбили на непересекающиеся слои $C_i = A_i \setminus A_{i+1}$ и на сердцевину $C = \bigcap_i A_i$.

Слои C_0, C_2, C_4, … равномощны (функция f осуществляет взаимно однозначное соответствие между C_0 и C_2, между C_2 и C_4 и т. д.):

$$C_0 \xrightarrow{f} C_2 \xrightarrow{f} C_4 \xrightarrow{f} \dots$$

То же самое можно сказать про слои с нечётными номерами:

$$C_1 \xrightarrow{f} C_3 \xrightarrow{f} C_5 \xrightarrow{f} \dots$$

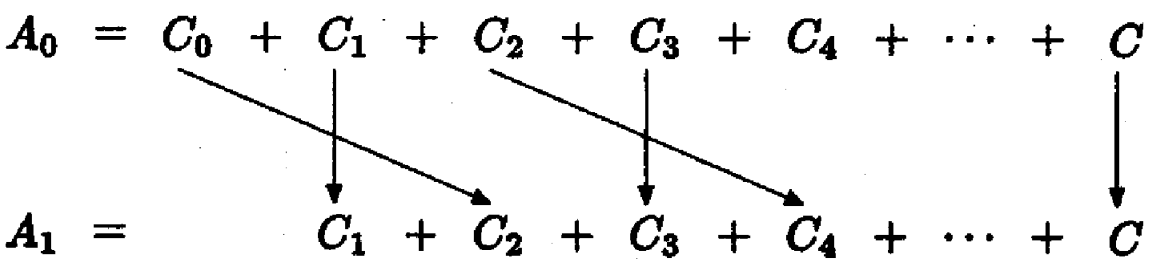


Рис. 4

Можно ещё отметить (что, впрочем, не понадобится), что функция f на множестве C осуществляет его перестановку (взаимно однозначное соответствие с самим собой).

Теперь легко понять, как построить взаимно однозначное соответствие g между A_0 и A_1. Пусть $x \in A_0$. Тогда соответствующий ему элемент $g(x)$ строится так: $g(x) = f(x)$ при $x \in C_{2k}$ и $g(x) = x$ при $x \in C_{2k+1}$ или $x \in C$ (см. рис. 4). □

Теорема Кантора – Бернштейна значительно упрощает доказательства равномощности: например, если мы хотим доказать, что бублик и шар в пространстве равномощны, то достаточно заметить, что из бублика можно вырезать маленький шар (гомотетичный большому), а из шара — маленький бублик.

Задача 42. *Посмотрите на приведённые выше задачи, где требовалось доказать равномощность, и убедитесь, что во многих из них применение теоремы Кантора – Бернштейна сильно упрощает дело.*

Задача 43. *Докажите, что все геометрические фигуры, содержащие хотя бы кусочек прямой или кривой, равномощны.*

Задача 44. *Докажите, что если квадрат разбит на два множества, то хотя бы одно из них равномощно квадрату. (Указание. Если одна из частей содержит отрезок, то можно воспользоваться теоремой Кантора – Бернштейна. Если же, скажем, первая часть не содержит отрезков,*

то в каждом горизонтальном сечении квадрата есть точка второй части, и снова можно сослаться на теорему Кантора – Бернштейна.)

Задача 45. *Докажите, что если отрезок разбит на две части, то хотя бы одна из них равномощна отрезку.*

То же самое доказательство можно изложить более абстрактно (и избавиться от упоминания натуральных чисел). Напомним, что $f\colon A \to A_2$ есть взаимно однозначное соответствие между множеством A и его подмножеством A_2, а A_1 — некоторое промежуточное множество. Назовём множество $X \subset A$ хорошим, если оно содержит $A \setminus A_1$ и замкнуто относительно f, т. е.

$$X \supset (A \setminus A_1) + f(X)$$

(мы используем знак + для объединения, поскольку объединяемые множества заведомо не пересекаются). Легко проверить, что пересечение любого семейства хороших множеств хорошее, поэтому если мы пересечём все хорошие множества, то получим минимальное по включению хорошее множество. Назовём его M. Легко проверить, что множество $(A \setminus A_1) + f(M)$ будет хорошим, поэтому в силу минимальности M включение в определении хорошего множества превращается в равенство:

$$M = (A \setminus A_1) + f(M).$$

Теперь всё готово для построения биекции $g\colon A \to A_1$. Эта биекция совпадает с f внутри M и тождественна вне M.

Задача 46. *Проведите это рассуждение подробно.*

Это рассуждение удобно при построении аксиоматической теории множеств, так как в нём не нужны натуральные числа (которые строятся далеко не сразу). Но по существу это то же самое рассуждение, поскольку M есть $C_0 \cup C_2 \cup \ldots$

Теперь, имея в виду теорему Кантора – Бернштейна, вернёмся к вопросу о сравнении мощностей. Для данных множеств A и B теоретически имеются четыре возможности:

- A равномощно некоторой части B, а B равномощно некоторой части A. (В этом случае, как мы знаем, множества равномощны.)
- A равномощно некоторой части B, но B не равномощно никакой части A. В этом случае говорят, что A *имеет меньшую мощность, чем* B.
- B равномощно некоторой части A, но A не равномощно никакой части B. В этом случае говорят, что A *имеет большую мощность, чем* B.

- Ни A не равномощно никакой части B, ни B не равномощно никакой части A. (Этот случай на самом деле невозможен).

Задача 47. *Докажите, что счётное множество имеет меньшую мощность, чем любое несчётное.*

Задача 48. *Проверьте аккуратно, что если A имеет меньшую мощность, чем B, а B имеет меньшую мощность, чем C, то A имеет меньшую мощность, чем C (транзитивность сравнения мощностей).*

Заметим, что мы уже долго говорим о сравнении мощностей, но воздерживаемся от упоминания мощности множества как самостоятельного объекта, а только сравниваем мощности разных множеств. В принципе можно было бы определить мощность множества A как класс всех множеств, равномощных A. Такие классы для множеств A и B совпадают в том и только том случае, когда A и B равномощны, так что слова имеют равную мощность приобрели бы буквальный смысл. Проблема тут в том, что таких множеств (равномощных множеству A) слишком много, поскольку всё на свете может быть их элементами. Их насколько много, что образовать из них множество затруднительно, это может привести к парадоксам (см. раздел 6, с. 28).

Из этой ситуации есть несколько выходов. Самый простой — по-прежнему говорить только о сравнении мощностей, но не о самих мощностях. Можно также ввести понятие класса — такой большой совокупности объектов, что её уже нельзя считать элементом других совокупностей (если вы понимаете, о чём я тут толкую), и считать мощностью множества A класс всех множеств, равномощных A. Ещё один выход — для каждого A выбрать некоторое стандартное множество, равномощное A, и назвать его мощностью множества A. Обычно в качестве стандартного множества берут минимальный ординал, равномощный A, — но это построение уже требует более формального (аксиоматического) построения теории множеств.

Так или иначе, мы будем употреблять обозначение $|A|$ для мощности множества A хотя бы как вольность речи: $|A| = |B|$ означает, что множества A и B равномощны; $|A| \leqslant |B|$ означает, что A равномощно некоторому подмножеству множества B, а $|A| < |B|$ означает, что A имеет меньшую мощность, чем B (см. с. 24).

§6. Теорема Кантора

Классический пример неравномощных бесконечных множеств (до сих пор такого примера у нас не было!) даёт диагональная конструкция Кантора.

Теорема 7 (Кантора). *Множество бесконечных последовательностей нулей и единиц несчётно.*

Доказательство. Предположим, что оно счётно. Тогда все последовательности нулей и единиц можно перенумеровать: $\alpha_0, \alpha_1, \ldots$ Составим бесконечную вниз таблицу, строками которой будут наши последовательности:

$$\begin{array}{lcccccc} \alpha_0 & = & \underline{\alpha_{00}} & \alpha_{01} & \alpha_{02} & \ldots \\ \alpha_1 & = & \alpha_{10} & \underline{\alpha_{11}} & \alpha_{12} & \ldots \\ \alpha_2 & = & \alpha_{20} & \alpha_{21} & \underline{\alpha_{22}} & \ldots \\ & \multicolumn{5}{c}{\ldots\ldots\ldots\ldots} \end{array}$$

(через α_{ij} мы обозначаем j-й член i-й последовательности). Теперь рассмотрим последовательность, образованную стоящими на диагонали членами α_{00}, α_{11}, α_{22}, $\ldots$; её i-й член есть α_{ii} и совпадает с i-м членом i-й последовательности. Заменив все члены на противоположные, мы получим последовательность β, у которой

$$\beta_i = 1 - \alpha_{ii},$$

так что последовательность β отличается от любой из последовательностей α_i (в позиции i) и потому отсутствует в таблице. А мы предположили, что таблица включает в себя все последовательности — противоречие. □

Из этой теоремы следует, что множество $\mathbb{R}$ действительных чисел (которое, как мы видели, равномощно множеству последовательностей нулей и единиц) несчётно. В частности, оно не может совпадать со счётным множеством алгебраических чисел и потому существует действительное число, не являющееся алгебраическим (не являющееся корнем никакого ненулевого многочлена с целочисленными коэффициентами). Такие числа называют *трансцендентными*.

К моменту создания Кантором теории множеств уже было известно, что такие числа существуют. Первый пример такого числа построил в 1844 году французский математик Ж. Лиувилль. Он показал, что число, хорошо приближаемое рациональными, не может быть алгебраическим (таково, например, число $\sum(1/10^{n!})$). Доказательство теоремы Лиувилля не очень сложно, но всё-таки требует некоторых оценок погрешности приближения; на его фоне доказательство Кантора, опубликованное им в 1874 году, выглядит чистой воды фокусом. Эта публикация была первой работой по теории множеств; в её первом параграфе доказывается счётность множества алгебраических чисел, а во втором — несчётность множества действительных чисел.

(Общее определение равномощности было дано Кантором лишь через три года, одновременно с доказательством равномощности пространств разного числа измерений, о котором мы уже говорили.)

Отметим кстати, что в том же 1874 году французский математик Ш. Эрмит доказал, что основание натуральных логарифмов e трансцендентно, а через восемь лет немецкий математик Ф. Линдеман доказал трансцендентность числа π и тем самым невозможности квадратуры круга.)

Задача 49. *Покажите, что любое замкнутое множество $A \subset \mathbb{R}$ либо конечно, либо счётно, либо имеет мощность континуума. (Указание. Рассмотрим множество $B \subset A$, состоящее из тех точек множества A, в любой окрестности которых несчётно много других точек из A. Если B пусто, то A конечно или счётно. Если B непусто, то оно замкнуто и не имеет изолированных точек.)*

Эта задача показывает, что для замкнутых подмножество прямой верна гипотеза континуума, утверждающая, что любое подмножество прямой либо конечно, либо счётно, либо равномощно $\mathbb{R}$. (Кантор, доказавший этот факт, рассматривал его как первый шаг к доказательству гипотезы континуума для общего случая, но из этого ничего не вышло.)

Вернёмся к диагональной конструкции. Мы знаем, что множество последовательностей нулей и единиц равномощно множеству подмножеств натурального ряда (каждому подмножеству соответствует его характеристическая последовательность, у которой единицы стоят на местах из этого подмножества). Поэтому можно переформулировать эту теорему так:

Множество $\mathbb{N}$ не равномощно множеству своих подмножеств.

Доказательство также можно шаг за шагом перевести на такой язык: пусть они равномощны; тогда есть взаимно однозначное соответствие $i \mapsto A_i$ между натуральными числами и подмножествами натурального ряда. Диагональная последовательность в этих терминах представляет собой множество тех i, для которых $i \in A_i$, а последовательность β, отсутствовавшая в перечислении, теперь будет его дополнением ($B = \{i \mid i \notin A_i\}$).

При этом оказывается несущественным, что мы имеем дело с натуральным рядом, и мы приходим к такому утверждению:

Теорема 8 (общая формулировка теоремы Кантора). *Никакое множество X не равномощно множеству всех своих подмножеств.*

Доказательство. Пусть φ — взаимно однозначное соответствие между X и множеством $P(X)$ всех подмножеств множества X. Рассмотрим те элемен-

ты $x \in X$, которые не принадлежат соответствующему им подмножеству. Пусть Z — образованное ими множество:

$$Z = \{x \in X \mid x \notin \varphi(x)\}.$$

Докажем, что подмножество Z не соответствует никакому элементу множества X. Пусть это не так и $Z = \varphi(z)$ для некоторого элемента $z \in X$. Тогда

$$z \in Z \Leftrightarrow z \notin \varphi(z) \Leftrightarrow z \notin Z$$

(первое — по построению множества Z, второе — по предположению $\varphi(z) = Z$). Полученное противоречие показывает, что Z действительно ничему не соответствует, так что φ не взаимно однозначно. □

С другой, стороны, любое множество X равномощно некоторой части множества $P(X)$. В самом деле, каждому элементу $x \in X$ можно поставить в соответствие одноэлементное подмножество $\{x\}$. Поэтому, вспоминая определение сравнения множеств по мощности (с. 24), можно сказать, что мощность множества X всегда меньше мощности множества $P(X)$

Задача 50. *Докажите, что $n < 2^n$ для всех натуральных $n = 0, 1, 2, \ldots$*

В доказательстве теоремы 8 Кантор вместо подмножеств рассуждал о функциях, принимающих значения 0 и 1.

На самом деле мы уже приблизились к опасной границе, когда наглядные представления о множествах приводят к противоречию. В самом деле, рассмотрим множество всех множеств U, элементами которого являются все множества. Тогда, в частности, все подмножества множества U будут его элементами, и $P(U) \subset U$, что невозможно по теореме Кантора.

Это рассуждение можно развернуть, вспомнив доказательство теоремы Кантора — получится так называемый парадокс Рассела. Вот как его обычно излагают.

Типичные множества не являются своими элементами. Скажем, множество натуральных чисел $\mathbb{N}$ само не является натуральным числом и потому не будет своим элементом. Однако в принципе можно себе представить и множество, которое является своим элементом (например, множество всех множеств). Назовём такие множества необычными. Рассмотрим теперь множество всех обычных множеств. Будет ли оно обычным? Если оно обычное, то оно является своим элементом и потому необычное, и наоборот. Как же так?

Модифицированная версия этого парадокса такова: будем называть прилагательное самоприменимым, если оно обладает описываемым свойством. Например, прилагательное русский самоприменимо, а прилагательное глиняный нет. Другой пример: прилагательное трёхсложный самоприменимо, а

двусложный нет. Теперь вопрос: будет ли прилагательное несамоприменимый самоприменимым? (Любой ответ очевидно приводит к противоречию.)

Отсюда недалеко до широко известного парадокса лжеца, говорящего я лгу, или до истории о солдате, который должен был брить всех солдат одной с ним части, кто не бреется сам и т. п.

Возвращаясь к теории множеств, мы обязаны дать себе отчёт в том, что плохого было в рассуждениях, приведших к парадоксу Рассела. Вопрос этот далеко не простой, и его обсуждение активно шло всю первую половину 20-го века. Итоги этого обсуждения приблизительно можно сформулировать так:

- Понятие множества не является непосредственно очевидным; разные люди (и научные традиции) могут понимать его по-разному.
- Множества — слишком абстрактные объекты для того, чтобы вопрос а что на самом деле? имел смысл. Например, в работе Кантора 1878 года была сформулирована *континуум-гипотеза*: всякое подмножество отрезка либо конечно, либо счётно, либо равномощно всему отрезку. (Другими словами, между счётными множествами и множествами мощности континуум нет промежуточных мощностей). Кантор написал, что это может быть доказано с помощью некоторого метода индукции, в изложение которого мы не будем входить здесь подробнее, но на самом деле доказать это ему не удалось. Более того, постепенно стало ясно, что утверждение континуум-гипотезы можно считать истинным или ложным, — при этом получаются разные теории множеств, но в общем-то ни одна из этих теорий не лучше другой.

  Тут есть некоторая аналогия с неевклидовой геометрией. Мы можем считать пятый постулат Евклида (через данную точку проходит не более одной прямой, параллельной данной) истинным. Тогда получится геометрия, называемая евклидовой. А можно принять в качестве аксиомы противоположное утверждение: через некоторую точку можно провести две различные прямые, параллельные некоторой прямой. Тогда получится неевклидова геометрия.

  Вопрос о том, евклидова или неевклидова геометрия правильна на самом деле, если вообще имеет смысл, не является математическим — скорее об этом следует спрашивать физиков. К теории множеств это относится в ещё большей степени, и разве что теология (Кантор неоднократно обсуждал вопросы теории множеств с профессионалами-теологами) могла бы в принципе претендовать на окончательный ответ.

- Если мы хотим рассуждать о множествах, не впадая в противоречия, нужно проявлять осторожность и избегать определённых видов рассуждений. Безопасные (по крайней мере пока не приведшие к противоречию) правила обращения со множествами сформулированы в аксиоматической теории множеств (формальная теория ZF, названная в честь Цермело и Френкеля). Добавив к этой теории аксиому выбора, получаем теорию, называемую ZFC (choice по-английски — выбор). Есть и другие, менее популярные теории.

Однако формальное построение теории множеств выходит за рамки нашего курса. Поэтому мы ограничимся неформальным описанием ограничений, накладываемых во избежание противоречий: нельзя просто так рассмотреть множество всех множеств или множество всех множеств, не являющихся своими элементами, поскольку класс потенциальных претендентов слишком необозрим. Множества можно строить лишь постепенно. Например, можно образовать множество всех подмножеств данного множества (*аксиома степени*). Можно рассмотреть подмножество данного множества, образованное элементами с каким-то свойством (*аксиома выделения*). Можно рассмотреть множество всех элементов, входящих хотя бы в один из элементов данного множества (*аксиома суммы*). Есть и другие аксиомы.

Излагая сведения из теории множеств, мы будем стараться держаться подальше от опасной черты, и указывать на опасность в тех местах, когда возникнет искушение к этой черте приблизиться. Пока что такое место было одно: попытка определить мощность множества как класс (множество) всех равномощных ему множеств.

§7. Функции

До сих пор мы старались ограничиваться минимумом формальностей и говорили о функциях, их аргументах, значениях, композиции и т. п. без попыток дать определения этих понятий. Сейчас мы дадим формальные определения.

Пусть A и B — два множества. Рассмотрим множество всех упорядоченных пар $\langle a, b\rangle$, где $a \in A$ и $b \in B$. Это множество называется *декартовым произведением* множеств A и B и обозначается $A \times B$. (К вопросу о том, что такое упорядоченная пара, мы ещё вернёмся на с. 34.)

Любое подмножество R множества $A \times B$ называется *отношением* между множествами A и B. Если $A = B$, говорят о *бинарном отношении* на множестве A. Например, на множестве натуральных чисел можно рассмотреть бинарное отношение быть делителем, обычно обозначаемое символом $|$. То-

гда можно в принципе было бы написать $\langle 2,6\rangle \in |$ и $\langle 2,7\rangle \notin |$. Обычно, однако, знак отношения пишут между объектами (например, $2|6$).

Задача 51. *Вопрос для самоконтроля: отношения быть делителем и делиться на — это одно и то же отношение или разные? (Ответ: конечно, разные — в упорядоченной паре порядок существен.)*

Если аргументами функции являются элементы множества A, а значениями — элементы множества B, то можно рассмотреть отношение между A и B, состоящее из пар вида $\langle x, f(x)\rangle$. По аналогии с графиками функций на плоскости такое множество можно назвать графиком функции f. С формальной точки зрения, однако, удобнее не вводить отдельного неопределяемого понятия функции, а вместо этого отождествить функцию с её графиком.

Отношение $F \subset A \times B$ называется *функцией из A в B*, если оно не содержит пар с одинаковым первым членом и разными вторыми. Другими словами, это означает, что для каждого $a \in A$ существует не более одного $b \in B$, при котором $\langle a, b\rangle \in F$.

Те элементы $a \in A$, для которых такое b существует, образуют *область определения* функции F. Она обозначается $D(F)$ (от английского слова domain). Для любого элемента $a \in D(F)$ можно определить *значение* функции F на аргументе a (в точке a, как иногда говорят) как тот единственный элемент $b \in B$, для которого $\langle a, b\rangle \in F$. Этот элемент записывают как $F(a)$. Все такие элементы b образуют *множество значений* функции F, которое обозначается $\mathrm{Im}(F)$.

Если $a \notin D(F)$, то говорят, что функция *не определена* на a. Заметим, что по нашему определению функция из A в B не обязана быть определена на всех элементах множества A — её область определения может быть любым подмножеством множества A. Симметричным образом множество её значений может не совпадать с множеством B.

Если область определения функции f из A в B совпадает с A, то пишут $f\colon A \to B$.

Пример: *тождественная* функция $\mathrm{id}_A\colon A \to A$ переводит множество A в себя, причём $\mathrm{id}(a) = a$ для любого $a \in A$. Она представляет собой множество пар вида $\langle a, a\rangle$ для всех $a \in A$. (Индекс A в id_A иногда опускают, если ясно, о каком множестве идёт речь.)

Композицией двух функций $f\colon A \to B$ и $g\colon B \to C$ называют функцию $h\colon A \to C$, определённую соотношением $h(x) = g(f(x))$. Другими словами, h представляет собой множество пар

$$\{\langle a, c\rangle \mid \langle a, b\rangle \in f \text{ и } \langle b, c\rangle \in g \text{ для некоторого } b \in B\}.$$

Композиция функций обозначается $g \circ f$ (мы, как и в большинстве книг, пишем справа функцию, которая применяется первой).

Очевидно, композиция (как операция над функциями) ассоциативна, то есть $h \circ (f \circ g) = (h \circ f) \circ g$, поэтому в композиции нескольких подряд идущих функций можно опускать скобки.

Пусть $f\colon A \to B$. *Прообразом* подмножества $B' \subset B$ называется множество всех элементов $x \in A$, для которых $f(x) \in B'$. Оно обозначается $f^{-1}(B')$:

$$f^{-1}(B') = \{x \in A \mid f(x) \in B'\}.$$

Образом множества $A' \subset A$ называется множество всех значений функции f на всех элементах множества A'. Оно обозначается $f(A')$:

$$\begin{aligned} f(A') &= \{f(a) \mid a \in A'\} = \\ &= \{b \in B \mid \langle a, b\rangle \in f \text{ для некоторого } a \in A'\}. \end{aligned}$$

Строго говоря, обозначение $f(A')$ может привести к путанице (одни и те же круглые скобки употребляются и для значения функции, и для образа множества), но обычно ясно, что имеется в виду.

Задача 52. *Какие из следующих равенств верны?*

$$\begin{aligned} f(A' \cap A'') &= f(A') \cap f(A''); \\ f(A' \cup A'') &= f(A') \cup f(A''); \\ f(A' \setminus A'') &= f(A') \setminus f(A''); \\ f^{-1}(B' \cap B'') &= f^{-1}(B') \cap f^{-1}(B''); \\ f^{-1}(B' \cup B'') &= f^{-1}(B') \cup f^{-1}(B''); \\ f^{-1}(B' \setminus B'') &= f^{-1}(B') \setminus f^{-1}(B''); \\ f^{-1}(f(A')) &\subset A'; \\ f^{-1}(f(A')) &\supset A'; \\ f(f^{-1}(B')) &\subset B'; \\ f(f^{-1}(B')) &\supset B'; \\ (g \circ f)(A) &= g(f(A)); \\ (g \circ f)^{-1}(C') &= f^{-1}(g^{-1}(C')); \end{aligned}$$

(Здесь $f\colon A \to B$, $g\colon B \to C$, $A', A'' \subset A$, $B', B'' \subset B$, $C' \subset C$.)

Иногда вместо функций говорят об отображениях (резервируя термин функция для отображений с числовыми аргументами и значениями). Мы

не будем строго придерживаться таких различий, употребляя слова отображение и функция как синонимы.

Функция $f\colon A \to B$ называется *инъективной*, или *инъекцией*, или *вложением*, если она переводит разные элементы в разные, то есть если $f(a_1) \neq \neq f(a_2)$ при различных a_1 и a_2.

Функция $f\colon A \to B$ называется *сюръективной*, или *сюръекцией*, или *наложением*, если множество её значений есть всё B. (Иногда такие функции называют *отображениями на* B.)

Эти два определения более симметричны, чем может показаться на первый взгляд, как показывают такие задачи:

Задача 53. *Докажите, что функция $f\colon A \to B$ является вложением тогда и только тогда, когда она имеет* левую обратную *функцию $g\colon B \to A$, то есть функцию g, для которой $g \circ f = \mathrm{id}_A$. Докажите, что функция $f\colon A \to B$ является наложением тогда и только тогда, когда она имеет* правую обратную *функцию $g\colon B \to A$, для которой $f \circ g = \mathrm{id}_B$.*

Задача 54. *Докажите, что функция $f\colon A \to B$ является вложением тогда и только тогда, когда на неё можно сокращать слева: из равенства $f \circ g_1 = f \circ g_2$ следует равенство $g_1 = g_2$ (для любых функций g_1, g_2, области значений которых содержатся в A). Докажите, что функция $f\colon A \to B$ является наложением тогда и только тогда, когда на неё можно сокращать справа: из равенства $g_1 \circ f = g_2 \circ f$ следует равенство $g_1 = g_2$ (для любых функций g_1, g_2, область определения которых есть B).*

Отображение (функция) $f\colon A \to B$, которое одновременно является инъекцией и сюръекцией (вложением и наложением), называется *биекцией*, или взаимно однозначным соответствием.

Если f — биекция, то существует *обратная* функция f^{-1}, для которой $f^{-1}(y) = x \Leftrightarrow f(x) = y$.

Задача 55. *Могут ли для некоторой функции левая и правая обратные существовать, но быть различны?*

Напомним, что множества A и B равномощны, если существует биекция $f\colon A \to B$. В каком случае существует инъекция (вложение) $f\colon A \to B$? Легко понять, что вложение является взаимно однозначным соответствием между A и некоторым подмножеством множества B, поэтому такое вложение существует тогда и только тогда, когда в B есть подмножество, равномощное A, т. е. когда мощность A не превосходит мощности B (в смысле определения, данного в разделе 5).

Чуть менее очевиден другой результат: наложение A на B существует тогда и только тогда, когда мощность B не превосходит мощности A.

В самом деле, пусть наложение $f\colon A \to B$ существует. Для каждого элемента $b \in B$ найдётся хотя бы один элемент $a \in A$, для которого $f(a) = b$. Выбрав по одному такому элементу, мы получим подмножество $A' \subset A$, которое находится во взаимно однозначном соответствии с множеством B. (Здесь снова используется аксиома выбора, о которой мы говорили на с. 15.)

В обратную сторону: если какое-то подмножество A' множества A равномощно множеству B и имеется биекция $g\colon A' \to B$, то наложение A на B можно получить, доопределив эту биекцию на элементах вне A' каким угодно образом.

Задача 56. *Найдите ошибку в этом рассуждении, не читая дальше.*

На самом деле такое продолжение возможно, только если B непусто, так что правильное утверждение звучит так: наложение A на B существует только и только тогда, когда B непусто и равномощно некоторому подмножеству A, или когда оба множества пусты.

В нашем изложении остаётся ещё один не вполне понятный момент: что такое упорядоченная пара? Неформально говоря, это способ из двух объектов x и y образовать один объект $\langle x, y\rangle$, причём этот способ обладает таким свойством:

$$\langle x_1, y_1\rangle = \langle x_2, y_2\rangle \;\Leftrightarrow\; x_1 = x_2 \text{ и } y_1 = y_2.$$

В принципе, можно так и считать понятие упорядоченной пары неопределяемым, а это свойство — аксиомой. Однако при формальном построении теории множеств удобно использовать трюк, придуманный польским математиком Куратовским, и избежать появления отдельного понятия упорядоченной пары. (Напомним, что $\{x\}$ обозначает множество, единственным элементом которого является x, а $\{x, y\}$ обозначает множество, которое содержит x и y и не содержит других элементов. Тем самым $\{x, y\} = \{x\} = \{y\}$, если $x = y$.)

Теорема 9 (Упорядоченная пара по Куратовскому). *Определим* $\langle x, y\rangle$ *как* $\{\{x\}, \{x, y\}\}$. *Тогда выполнено указанное выше свойство:*

$$\langle x_1, y_1\rangle = \langle x_2, y_2\rangle \;\Leftrightarrow\; x_1 = x_2 \text{ и } y_1 = y_2.$$

Доказательство. Пусть $\langle x_1, y_1\rangle = \langle x_2, y_2\rangle$. По определению это означает, что

$$\{\{x_1\}, \{x_1, y_1\}\} = \{\{x_2\}, \{x_2, y_2\}\}.$$

Теперь нужно аккуратно разобрать случаи (не путая при этом x с $\{x\}$). Это удобно делать в следующем порядке. Пусть сначала $x_1 \neq y_1$. Тогда множество $\{x_1, y_1\}$ состоит из двух элементов. Раз оно принадлежит левой части равенства, то принадлежит и правой. Значит, оно равно либо $\{x_2\}$,

либо $\{x_2, y_2\}$. Первое невозможно, так как двухэлементное множество не может быть равно одноэлементному. Значит, $\{x_1, y_1\} = \{x_2, y_2\}$. С другой стороны, одноэлементное множество $\{x_1\}$ принадлежит левой части равенства, поэтому оно принадлежит и правой, и потому равно $\{x_2\}$ (поскольку не может быть равно двухэлементному). Отсюда $x_1 = x_2$ и $y_1 = y_2$, что и требовалось.

Аналогично можно разобрать симметричный случай, когда $x_2 \neq y_2$.

Осталось рассмотреть ситуацию, когда $x_1 = y_1$ и $x_2 = y_2$. В этом случае $\{x_1, y_1\} = \{x_1\}$ и потому левая часть данного нам равенства есть $\{\{x_1\}\}$. Аналогичным образом, правая его часть есть $\{\{x_2\}\}$, и потому $x_1 = x_2$, так что все четыре элемента x_1, x_2, y_1, y_2 совпадают. □

Заметим, что возможны и другие определения упорядоченной пары, для которых аналогичное утверждение верно, так что никакого философского смысла в этом определении нет — это просто удобный технический приём.

Задача 57. *Докажите утверждение теоремы 9 для* упорядоченной пары по Винеру*:* $\langle x, y\rangle = \{\{\varnothing, \{x\}\}, \{\{y\}\}\}$.

ГЛАВА II

Упорядоченные множества

§1. Отношения эквивалентности и порядка

Напомним, что бинарным отношением на множестве X называется подмножество $R \subset X \times X$; вместо $\langle x_1, x_2 \rangle \in R$ часто пишут $x_1 R x_2$.

Бинарное отношение R на множестве X называется *отношением эквивалентности*, если выполнены следующие свойства:

- (рефлексивность) xRx для всех $x \in X$;
- (симметричность) $xRy \Rightarrow yRx$ для всех $x, y \in X$;
- (транзитивность) xRy и $yRz \Rightarrow xRz$ для любых элементов $x, y, z \in \in X$.

Имеет место следующее очевидное, но часто используемое утверждение:

Теорема 10. *(а) Если множество X разбито в объединение непересекающихся подмножеств, то отношение лежать в одном подмножестве является отношением эквивалентности.*

(б) Всякое отношение эквивалентности получается описанным способом из некоторого разбиения.

Доказательство. Первое утверждение совсем очевидно; мы приведём доказательство второго, чтобы было видно, где используются все пункты определения эквивалентности. Итак, пусть R — отношение эквивалентности. Для каждого элемента $x \in X$ рассмотрим его *класс эквивалентности* — множество всех $y \in X$, для которых верно xRy.

Докажем, что для двух различных x_1, x_2 такие множества либо не пересекаются, либо совпадают. Пусть они пересекаются, то есть имеют общий элемент z. Тогда $x_1 Rz$ и $x_2 Rz$, откуда zRx_2 (симметричность) и $x_1 Rx_2$ (транзитивность), а также $x_2 Rx_1$ (симметричность). Поэтому для любого z из $x_1 Rz$ следует $x_2 Rz$ (транзитивность) и наоборот.

Осталось заметить, что в силу рефлексивности каждый элемент x принадлежит задаваемому им классу, то есть действительно всё множество X разбито на непересекающиеся классы. □

Задача 58. *Покажите, что требования симметричности и транзитивности можно заменить одним: xRz и $yRz \Rightarrow xRy$ (при сохранении требования рефлексивности).*

Задача 59. *Сколько различных отношений эквивалентности существует на множестве* $\{1, 2, 3, 4, 5\}$*?*

Задача 60. *(Теорема Рамсея) Множество всех* k*-элементных подмножеств бесконечного множества* A *разбито на* l *классов (*k*,* l *— натуральные числа). Докажите, что найдётся бесконечное множество* $B \subset A$*, все* k*-элементные подмножества которого принадлежат одному классу.*

(При $k = 1$ *это очевидно: если бесконечное множество разбито на конечное число классов, то один из классов бесконечен. При* $k = 2$ *и* $l = 2$ *утверждение можно сформулировать так: из бесконечного множества людей можно выбрать либо бесконечно много попарно знакомых, либо бесконечно много попарно незнакомых. Конечный вариант этого утверждения — о том, что среди любых шести людей есть либо три попарно знакомых, либо три попарно незнакомых, — известная задача для школьников.)*

Множество классов эквивалентности называют *фактор-множеством* множества X по отношению эквивалентности R. (Если отношение согласовано с дополнительными структурами на X, получаются фактор-группы, фактор-кольца и т. д.)

Отношения эквивалентности нам не раз ещё встретятся, но сейчас наша основная тема — отношения порядка.

Бинарное отношение $\leqslant$ на множестве X называется *отношением частичного порядка*, если выполнены такие свойства:

- (рефлексивность) $x \leqslant x$ для всех $x \in X$;
- (антисимметричность) $x \leqslant y$ и $y \leqslant x \Rightarrow x = y$ для всех $x, y \in X$;
- (транзитивность) $x \leqslant y$ и $y \leqslant z \Rightarrow x \leqslant z$ для всех $x, y, z \in X$.

(Следуя традиции, мы используем символ $\leqslant$ (а не букву) как знак отношения порядка.) Множество с заданным на нём отношением частичного порядка называют *частично упорядоченным*.

Говорят, что два элемента x, y частично упорядоченного множества *сравнимы*, если $x \leqslant y$ или $y \leqslant x$. Заметим, что определение частичного порядка не требует, чтобы любые два элемента множества были сравнимы. Добавив это требование, мы получим определение *линейного порядка* (*линейно упорядоченного множества*).

Приведём несколько примеров частичных порядков:

- Числовые множества с обычным отношением порядка (здесь порядок будет линейным).
- На множестве $\mathbb{R} \times \mathbb{R}$ всех пар действительных чисел можно ввести частичный порядок, считая, что $\langle x_1, x_2\rangle \leqslant \langle y_1, y_2\rangle$, если $x_1 \leqslant x_2$ и $y_1 \leqslant$

$\leqslant y_2$. Этот порядок уже не будет линейным: пары $\langle 0,1\rangle$ и $\langle 1,0\rangle$ не сравнимы.

- На множестве функций с действительными аргументами и значениями можно ввести частичный порядок, считая, что $f \leqslant g$, если $f(x) \leqslant g(x)$ при всех $x \in \mathbb{R}$. Этот порядок не будет линейным.
- На множестве целых положительных чисел можно определить порядок, считая, что $x \leqslant y$, если x делит y. Этот порядок тоже не будет линейным.
- Отношение любой простой делитель числа x является также и делителем числа y не будет отношением порядка на множестве целых положительных чисел (оно рефлексивно и транзитивно, но не антисимметрично).
- Пусть U — произвольное множество. Тогда на множестве $P(U)$ всех подмножеств множества U отношение включения $\subset$ будет частичным порядком.
- На буквах русского алфавита традиция определяет некоторый порядок (а $\leqslant$ б $\leqslant$ в $\leqslant \ldots \leqslant$ я). Этот порядок линеен — про любые две буквы можно сказать, какая из них раньше (при необходимости заглянув в словарь).
- На словах русского алфавита определён *лексикографический* порядок (как в словаре). Формально определить его можно так: если слово x является началом слова y, то $x \leqslant y$ (например, кант $\leqslant$ кантор). Если ни одно из слов не является началом другого, посмотрим на первую по порядку букву, в которой слова отличаются: то слово, где эта буква меньше в алфавитном порядке, и будет меньше. Этот порядок также линеен (иначе что бы делали составители словарей?).
- Отношение равенства ($(x \leqslant y) \Leftrightarrow (x = y)$) также является отношением частичного порядка, для которого никакие два различных элемента не сравнимы.
- Приведём теперь бытовой пример. Пусть есть множество X картонных коробок. Введём на нём порядок, считая, что $x \leqslant y$, если коробка x целиком помещается внутрь коробки y (или если x и y — одна и та же коробка). В зависимости от набора коробок этот порядок может быть или не быть линейным.

Пусть x, y — элементы частично упорядоченного множества X. Говорят, что $x < y$, если $x \leqslant y$ и $x \neq y$. Для этого отношения выполнены такие

свойства:

$$x \not< x;$$
$$(x < y) \text{ и } (y < z) \;\Rightarrow\; x < z.$$

(Первое очевидно, проверим второе: если $x < y$ и $y < z$, то есть $x \leqslant y$, $x \neq y$, $y \leqslant z$, $y \neq z$, то $x \leqslant z$ по транзитивности; если бы оказалось, что $x = z$, то мы бы имели $x \leqslant y \leqslant x$ и потому $x = y$ по антисимметричности, что противоречит предположению.)

Терминологическое замечание: мы читаем знак $\leqslant$ как меньше или равно, а знак $<$ — как меньше, неявно предполагая, что $x \leqslant y$ тогда и только тогда, когда $x < y$ или $x = y$. К счастью, это действительно так. Ещё одно замечание: выражение $x > y$ (x больше y) означает, что $y < x$, а выражение $x \geqslant y$ (x больше или равно y) означает, что $y \leqslant x$.

Задача 61. *Объясните, почему не стоит читать $x \leqslant y$ как x не больше y.*

В некоторых книжках отношение частичного порядка определяется как отношение $<$, удовлетворяющее двум указанным свойствам. В этом случае отношение $x \leqslant y \Leftrightarrow [(x < y) \text{ или } (x = y)]$ является отношением частичного порядка в смысле нашего определения.

Задача 62. *Проверьте это.*

Во избежание путаницы отношение $<$ иногда называют отношением *строгого порядка*, а отношение $\leqslant$ — отношением *нестрогого порядка*. Одно и то же частично упорядоченное множество можно задавать по-разному: можно сначала определить отношение нестрогого порядка $\leqslant$ (рефлексивное, антисимметричное и транзитивное) и затем из него получить отношение строгого порядка $<$, а можно действовать и наоборот.

Задача 63. *Опуская требование антисимметричности в определении частичного порядка, получаем определение* предпорядка. *Докажите, что любой предпорядок устроен так: множество делится на непересекающиеся классы, при этом $x \leqslant y$ для любых двух элементов x, y из одного класса, а на фактор-множестве задан частичный порядок, который и определяет результат сравнения двух элементов из разных классов.*

Вот несколько конструкций, позволяющих строить одни упорядоченные множества из других.

- Пусть Y — подмножество частично упорядоченного множества $(X, \leqslant)$. Тогда на множестве Y возникает естественный частичный порядок, *индуцированный* из X. Формально говоря,

$$(\leqslant_Y) = (\leqslant) \cap (Y \times Y).$$

  Если порядок на X был линейным, то и индуцированный порядок на Y, очевидно, будет линейным.
- Пусть X и Y — два непересекающихся частично упорядоченных множества. Тогда на их объединении можно определить частичный порядок так: внутри каждого множества элементы сравниваются как раньше, а любой элемент множества X по определению меньше любого элемента Y. Это множество естественно обозначить $X+Y$. (Порядок будет линейным, если он был таковым на каждом из множеств.)

  Это же обозначение применяют и для пересекающихся (и даже совпадающих множеств). Например, говоря об упорядоченном множестве $\mathbb{N}+\mathbb{N}$, мы берём для непересекающиеся копии натурального ряда $\{0, 1, 2, \dots\}$ и $\{\overline{0}, \overline{1}, \overline{2} \dots\}$ и рассматриваем множество $\{0, 1, 2, \dots, \overline{0}, \overline{1}, \overline{2}, \dots\}$, причём $k \leqslant \overline{l}$ при всех k и l, а внутри каждой копии порядок обычный.
- Пусть $(X, \leqslant_X)$ и $(Y, \leqslant_Y)$ — два упорядоченных множества. Можно определить порядок на произведении $X \times Y$ несколькими способами. Можно считать, что $\langle x_1, y_1\rangle \leqslant \langle x_2, y_2\rangle$, если $x_1 \leqslant_X x_2$ и $y_1 \leqslant_Y y_2$ (покоординатное сравнение). Этот порядок, однако, не будет линейным, даже если исходные порядки и были линейными: если первая координата больше у одной пары, а вторая у другой, как их сравнить? Чтобы получить линейный порядок, договоримся, какая координата будет главной и будем сначала сравнивать по ней, а потом (в случае равенства) — по другой. Если главной считать X-координату, то $\langle x_1, y_1\rangle \leqslant \langle x_2, y_2\rangle$, если $x_1 <_X x_2$ или если $x_1 = x_2$, а $y_1 \leqslant_Y y_2$. Однако по техническим причинам удобно считать главной вторую координату. Говоря о произведении двух линейно упорядоченных множеств как о линейно упорядоченном множестве, мы в дальнейшем подразумеваем именно такой порядок (сначала сравниваем по второй координате).

Задача 64. *Докажите, что в частично упорядоченном множестве $\mathbb{N} \times \mathbb{N}$ (порядок покоординатный) нет бесконечного подмножества, любые два элемента которого были бы несравнимы. Верно ли аналогичное утверждение для $\mathbb{Z} \times \mathbb{Z}$?*

Задача 65. *Докажите аналогичное утверждение для $\mathbb{N}^k$ (порядок покоординатный).*

Задача 66. *Пусть U — конечное множество из n элементов. Рассмотрим множество $P(U)$ всех подмножеств множества U, упорядоченное по включению. Какова максимально возможная мощность множества $S \subset P(U)$, если индуцированный на S порядок линеен? если никакие два элемента S не сравнимы? (Указание: см. задачу 12.)*

Задача 67. *Сколько существует различных линейных порядков на множестве из n элементов?*

Задача 68. *Докажите, что всякий частичный порядок на конечном множестве можно продолжить до линейного (продолжить означает, что если $x \leqslant y$ в исходном порядке, то и в новом это останется так).*

Задача 69. *Дано бесконечное частично упорядоченное множество X. Докажите, что в нём всегда найдётся либо бесконечное подмножество попарно несравнимых элементов, либо бесконечное подмножество, на котором индуцированный порядок линеен.*

Задача 70. *(Конечный вариант предыдущей задачи.) Даны целые положительные числа m и n. Докажите, что во всяком частично упорядоченном множестве мощности $mn+1$ можно указать либо $m+1$ попарно несравнимых элементов, либо $n+1$ попарно сравнимых.*

Задача 71. *В строчку написаны $mn+1$ различных чисел. Докажите, что можно часть из них вычеркнуть так, чтобы осталась либо возрастающая последовательность длины $m+1$, либо убывающая последовательность длины $n+1$. (Указание: можно воспользоваться предыдущей задачей.)*

Задача 72. *Рассмотрим семейство всех подмножеств натурального ряда, упорядоченное по включению. Существует ли у него линейно упорядоченное (в индуцированном порядке) подсемейство мощности континуум? Существует ли у него подсемейство мощности континуум, любые два элемента которого несравнимы?*

Элемент частично упорядоченного множества называют *наибольшим*, если он больше любого другого элемента, и *максимальным*, если не существует большего элемента. Если множество не является линейно упорядоченным, то это не одно и то же: наибольший элемент автоматически является максимальным, но не наоборот. (Одно дело коробка, в которую помещается любая другая, другое — коробка, которая никуда больше не помещается.)

Аналогичным образом определяются *наименьшие* и *минимальные* элементы.

Легко понять, что наибольший элемент в данном частично упорядоченном множестве может быть только один, в то время как максимальных элементов может быть много.

Задача 73. *Докажите, что любые два максимальных элемента не сравнимы. Докажите, что в конечном частично упорядоченном множестве X для любого элемента x найдётся максимальный элемент y, больший или равный x.*

§2. Изоморфизмы

Два частично упорядоченных множества называются *изоморфными*, если между ними существует *изоморфизм*, то есть взаимно однозначное соответствие, сохраняющее порядок. (Естественно, что в этом случае они равномощны как множества.) Можно сказать так: биекция $f\colon A \to B$ называется изоморфизмом частично упорядоченных множеств A и B, если

$$a_1 \leqslant a_2 \;\Leftrightarrow\; f(a_1) \leqslant f(a_2)$$

для любых элементов $a_1, a_2 \in A$ (слева знак $\leqslant$ обозначает порядок в множестве A, справа — в множестве B).

Очевидно, что отношение изоморфности рефлексивно (каждое множество изоморфно самому себе), симметрично (если X изоморфно Y, то и наоборот) и транзитивно (два множества, изоморфные третьему, изоморфны между собой). Таким образом, все частично упорядоченные множества разбиваются на классы изоморфных, которые называют *порядковыми типами*. (Правда, как и с мощностями, тут необходима осторожность — изоморфных множеств слишком много, и потому говорить о порядковых типах как множествах нельзя.)

Теорема 11. *Конечные линейно упорядоченные множества из одинакового числа элементов изоморфны.*

Доказательство. Конечное линейно упорядоченное множество всегда имеет наименьший элемент (возьмём любой элемент; если он не наименьший, возьмём меньший, если и он не наименьший, ещё меньший — и так далее; получим убывающую последовательность $x > y > z > \ldots$, которая рано или поздно должна оборваться). Присвоим наименьшему элементу номер 1. Из оставшихся снова выберем наименьший элемент и присвоим ему

номер 2 и так далее. Легко понять, что порядок между элементами соответствует порядку между номерами, то есть что наше множество изоморфно множеству $\{1, 2, \dots, n\}$. □

Задача 74. *Докажите, что множество всех целых положительных делителей числа* 30 *с отношением быть делителем в качестве отношения порядка изоморфно множеству всех подмножеств множества* $\{a, b, c\}$, *упорядоченному по включению.*

Задача 75. *Будем рассматривать финитные последовательности натуральных чисел, то есть последовательности, у которых все члены, кроме конечного числа, равны* 0. *На множестве таких последовательностей введём покомпонентный порядок:* $(a_0, a_1, \dots) \leqslant (b_0, b_1, \dots)$, *если* $a_i \leqslant b_i$ *при всех* i. *Докажите, что это множество изоморфно множеству всех положительных целых чисел с отношением быть делителем в качестве порядка.*

Взаимно однозначное отображение частично упорядоченного множества A в себя, являющееся изоморфизмом, называют *автоморфизмом* частично упорядоченного множества A. Тождественное отображение всегда является автоморфизмом, но для некоторых множеств существуют и другие автоморфизмы. Например, отображение прибавления единицы ($x \mapsto x + 1$) является автоморфизмом частично упорядоченного множества $\mathbb{Z}$ целых чисел (с естественным порядком). Для множества натуральных чисел та же формула не даёт автоморфизма (нет взаимной однозначности).

Задача 76. *Покажите, что не существует автоморфизма упорядоченного множества* $\mathbb{N}$ *натуральных чисел, отличного от тождественного.*

Задача 77. *Рассмотрим множество* $P(A)$ *всех подмножеств некоторого* k*-элементного множества* A, *частично упорядоченное по включению. Найдите число автоморфизмов этого множества.*

Задача 78. *Покажите, что множество целых положительных чисел, частично упорядоченное отношением* x *делит* y, *имеет континуум различных автоморфизмов.*

Вот несколько примеров равномощных, но не изоморфных линейно упорядоченных множеств (в силу теоремы 11 они должны быть бесконечными).

- Отрезок $[0, 1]$ (с обычным отношением порядка) не изоморфен множеству $\mathbb{R}$, так как у первого есть наибольший элемент, а у второго нет.

(При изоморфизме наибольший элемент, естественно, должен соответствовать наибольшему.)

- Множество $\mathbb{Z}$ (целые числа с обычным порядком) не изоморфно множеству $\mathbb{Q}$ (рациональные числа). В самом деле, пусть $\alpha\colon \mathbb{Z} \to \mathbb{Q}$ является изоморфизмом. Возьмём два соседних целых числа, скажем, 2 и 3. При изоморфизме α им должны соответствовать какие-то два рациональных числа $\alpha(2)$ и $\alpha(3)$, причём $\alpha(2) < \alpha(3)$, так как $2 < 3$. Но тогда рациональным числам между $\alpha(2)$ и $\alpha(3)$ должны соответствовать целые числа между 2 и 3, которых нет.
- Более сложный пример — множества $\mathbb{Z}$ и $\mathbb{Z} + \mathbb{Z}$. Возьмём в $\mathbb{Z} + \mathbb{Z}$ две копии нуля (из той и другой компоненты); мы обозначали их 0 и $\overline{0}$. При этом $0 < \overline{0}$. При изоморфизме им должны соответствовать два целых числа a и b, для которых $a < b$. Тогда всем элементам между 0 и $\overline{0}$ (их бесконечно много: 1, 2, 3, ..., $-\overline{3}$, $-\overline{2}$, $-\overline{1}$) должны соответствовать числа между a и b — но их лишь конечное число.

  Этот пример принципиально отличается от предыдущих тем, что здесь разницу между свойствами множеств нельзя записать формулой. Как говорят, упорядоченные множества $\mathbb{Z}$ и $\mathbb{Z} + \mathbb{Z}$ элементарно эквивалентны.

Задача 79. *Докажите, что линейно упорядоченные множества $\mathbb{Z} \times \mathbb{N}$ и $\mathbb{Z} \times \mathbb{Z}$ (с описанным выше на с. 40 порядком) не изоморфны.*

Задача 80. *Будут ли изоморфны линейно упорядоченные множества $\mathbb{N} \times \mathbb{Z}$ и $\mathbb{Z} \times \mathbb{Z}$?*

Задача 81. *Будут ли изоморфны линейно упорядоченные множества $\mathbb{Q} \times \mathbb{Z}$ и $\mathbb{Q} \times \mathbb{N}$?*

Отображение $x \mapsto \sqrt{2}x$ осуществляет изоморфизм между интервалами $(0, 1)$ и $(0, \sqrt{2})$. Но уже не так просто построить изоморфизм между множествами рациональных точек этих интервалов (то есть между $\mathbb{Q} \cap (0, 1)$ и $\mathbb{Q} \cap (0, \sqrt{2})$), поскольку умножение на $\sqrt{2}$ переводит рациональные числа в иррациональные. Тем не менее изоморфизм построить можно. Для этого надо взять возрастающие последовательности рациональных чисел $0 < x_1 < x_2 < \ldots$ и $0 < y_1 < y_2 < \ldots$, сходящиеся соответственно к 1 и $\sqrt{2}$ и построить кусочно-линейную функцию f, которая переводит x_i в y_i и линейна на каждом из отрезков $[x_i, x_{i+1}]$ (рис. 1). Легко понять, что она будет искомым изоморфизмом.

Задача 82. *Покажите, что множество рациональных чисел интервала $(0, 1)$ и множество $\mathbb{Q}$ изоморфны. (Указание: здесь тоже можно построить ломаную; впрочем, у этой задачи есть и другое решение, которое*

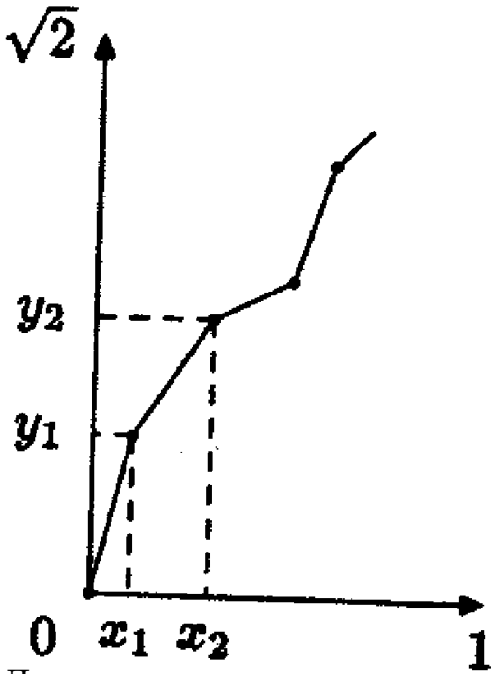


Рис. 1. Ломаная осуществляет изоморфизм.

начинается с того, что функция $x \mapsto 1/x$ переводит рациональные числа в рациональные.)

Более сложная конструкция требуется в следующей задаче (видимо, ничего проще, чем сослаться на общую теорему 12, тут не придумаешь).

Задача 83. *Докажите, что множество двоично-рациональных чисел интервала $(0,1)$ изоморфно множеству $\mathbb{Q}$. (Число считается* двоично-рациональным*, если оно имеет вид $m/2^n$, где m — целое число, а n — натуральное.)*

Два элемента x, y линейно упорядоченного множества называют *соседними*, если $x < y$ и не существует элемента между ними, то есть такого z, что $x < z < y$. Линейно упорядоченное множество называют *плотным*, если в нём нет соседних элементов (то есть между любыми двумя есть третий).

Теорема 12. *Любые два счётных плотных линейно упорядоченных множества без наибольшего и наименьшего элементов изоморфны.*

Доказательство. Пусть X и Y — данные нам множества. Требуемый изоморфизм между ними строится по шагам. После n шагов у нас есть два n-элементных подмножества $X_n \subset X$ и $Y_n \subset Y$, элементы которых мы будем называть охваченными, и взаимно однозначное соответствие между ними, сохраняющее порядок. На очередном шаге мы берём какой-то неохваченный элемент одного из множеств (скажем, множества X) и сравниваем

его со всеми охваченными элементами X. Он может оказаться либо меньше всех, либо больше, либо попасть между какими-то двумя. В каждом из случаев мы можем найти неохваченный элемент в Y, находящийся в том же положении (больше всех, между первым и вторым охваченным сверху, между вторым и третьим охваченным сверху и т. п.). При этом мы пользуемся тем, что в Y нет наименьшего элемента, нет наибольшего и нет соседних элементов, — в зависимости от того, какой из трёх случаев имеет место. После этого мы добавляем выбранные элементы к X_n и Y_n, считая их соответствующими друг другу.

Чтобы в пределе получить изоморфизм между множествами X и Y, мы должны позаботиться о том, чтобы все элементы обоих множеств были рано или поздно охвачены. Это можно сделать так: поскольку каждое из множеств счётно, пронумеруем его элементы и будем выбирать неохваченный элемент с наименьшим номером (на нечётных шагах — из X, на чётных — из Y). Это соображение завершает доказательство. □

Задача 84. *Сколько существует неизоморфных счётных плотных линейно упорядоченных множеств (про наименьший и наибольший элементы ничего не известно). (Ответ:* 4*.)*

Задача 85. *Приведите пример двух плотных линейно упорядоченных множеств мощности континуум без наименьшего и наибольшего элементов, не являющихся изоморфными. (Указание: возьмите множества* $\mathbb{Q}+\mathbb{R}$ *и* $\mathbb{R}+\mathbb{Q}$*.)*

Теорема 13. *Всякое счётное линейно упорядоченное множество изоморфно некоторому подмножеству множества* $\mathbb{Q}$*.*

Доказательство. Заметим сразу же, что вместо множества $\mathbb{Q}$ можно было взять любое плотное счётное всюду плотное множество без первого и последнего элементов, так как они все изоморфны.

Доказательство этого утверждения происходит так же, как и в теореме 12 — с той разницей, что новые необработанные элементы берутся только с одной стороны (из данного нам множества), а пары к ним подбираются в множестве рациональных чисел. □

§3. Вполне упорядоченные множества

Принцип математической индукции в одной из возможных форм звучит так:

Пусть $A(n)$ — некоторое свойство натурального числа n. Пусть нам удалось доказать $A(n)$ в предположении, что $A(m)$ верно для всех m, меньших n. Тогда свойство $A(n)$ верно для всех натуральных чисел n.

(Заметим, что по условию доказательство $A(0)$ возможно без всяких предположений, поскольку меньших чисел нет.)

Для каких частично упорядоченных множеств верен аналогичный принцип? Ответ даётся следующей простой теоремой:

Теорема 14. *Следующие три свойства частично упорядоченного множества X равносильны:*

(а) любое непустое подмножество X имеет минимальный элемент;

(б) не существует бесконечной строго убывающей последовательности $x_0 > x_1 > x_2 > \ldots$ элементов множества X;

(в) для множества X верен принцип индукции в следующей форме: если (при каждом $x \in X$) из истинности $A(y)$ для всех $y < x$ следует истинность $A(x)$, то свойство $A(x)$ верно при всех x. Формально это записывают так:

$$\forall x\,(\forall y\,((y < x) \Rightarrow A(y)) \Rightarrow A(x)) \Rightarrow \forall x\,A(x).$$

Доказательство. Сперва докажем эквивалентность первых двух свойств. Если $x_0 > x_1 > x_2 > \ldots$ — бесконечная убывающая последовательность, то, очевидно, множество её значений не имеет минимального элемента (для каждого элемента следующий ещё меньше). Поэтому из (а) следует (б). Напротив, если B — непустое множество, не имеющее минимального элемента, то бесконечную убывающую последовательность можно построить так. Возьмём произвольный элемент $b_0 \in B$. По предположению он не является минимальным, так что можно найти $b_1 \in B$, для которого $b_0 > b_1$. По тем же причинам можно найти $b_2 \in B$, для которого $b_1 > b_2$ и т. д. Получается бесконечная убывающая последовательность.

Теперь выведем принцип индукции из существования минимального элемента в любом подмножестве. Пусть $A(x)$ — произвольное свойство элементов множества X, верное не для всех элементов x. Рассмотрим непустое множество B тех элементов, для которых свойство A неверно. Пусть x — минимальный элемент множества B. По условию меньших элементов в множестве B нет, поэтому для всех $y < x$ свойство $A(y)$ выполнено. Но тогда по предположению должно быть выполнено и $A(x)$ — противоречие.

Осталось доказать существование минимального элемента в любом непустом подмножестве, исходя из принципа индукции. Пусть B — подмножество без минимальных элементов. Докажем по индукции, что B пусто; другими словами, в качестве $A(x)$ возьмём свойство $x \notin B$. В самом деле,

если $A(y)$ верно для всех $y < x$, то никакой элемент, меньший x, не лежит в B. Если бы x лежал в B, то он был бы там минимальным, а таких нет. □

Множества, обладающие свойствами (а)–(в), называются *фундированными*. Какие есть примеры фундированных множеств? Прежде всего, наш исходный пример — множество натуральных чисел.

Другой пример — множество $\mathbb{N} \times \mathbb{N}$ пар натуральных чисел (меньше та пара, у которой второй член меньше; в случае равенства сравниваем первые). В самом деле, проверим условие (б). Нам будет удобно сформулировать его так: всякая последовательность $u_0 \geqslant u_1 \geqslant u_2 \geqslant \ldots$ элементов множества рано или поздно стабилизируется (все члены, начиная с некоторого, равны); очевидно, что это эквивалентная формулировка.

Пусть дана произвольная последовательность пар

$$\langle x_0, y_0\rangle \geqslant \langle x_1, y_1\rangle \geqslant \langle x_2, y_2\rangle \geqslant \ldots$$

По определению порядка (сначала сравниваются вторые члены) $y_0 \geqslant y_1 \geqslant y_2 \geqslant \ldots$ и потому последовательность натуральных чисел y_i с какого-то места не меняется. После этого уже x_i должны убывать — и тоже стабилизируются. Что и требовалось.

То же самое рассуждение пригодно и в более общей ситуации.

Теорема 15. *Пусть A и B — два фундированных частично упорядоченных множества. Тогда их произведение $A \times B$, в котором*

$$\langle a_1, b_1\rangle \leqslant \langle a_2, b_2\rangle \Leftrightarrow [(b_1 < b_2) \text{ или } (b_1 = b_2 \text{ и } a_1 \leqslant a_2)],$$

является фундированным.

Доказательство. В последовательности $\langle a_0, b_0\rangle \geqslant \langle a_1, b_1\rangle \geqslant \ldots$ стабилизируются сначала вторые, а затем и первые члены. □

Отсюда вытекает аналогичное утверждение для $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$, для $\mathbb{N}^k$ или вообще для произведения конечного числа фундированных множеств.

Ещё проще доказать, что сумма $A + B$ двух фундированных множеств A и B фундирована: последовательность $x_0 \leqslant x_1 \leqslant x_2 \leqslant \ldots$ либо целиком содержится в B (и мы ссылаемся на фундированность B), либо содержит элемент из A. В последнем случае все следующие элементы также принадлежат A, и мы используем фундированность A.

Часто в программировании (или в олимпиадных задачах) нам нужно доказать, что некоторый процесс не может продолжаться бесконечно долго. Например, написав цикл, мы должны убедиться, что рано или поздно из него выйдем. Это можно сделать так: ввести какой-то натуральный параметр и убедиться, что на каждом шаге цикла этот параметр уменьшается.

Тогда, если сейчас этот параметр равен N, то можно гарантировать, что не позже чем через N шагов цикл закончится.

Однако бывают ситуации, в которых число шагов заранее оценить нельзя, но тем не менее гарантировать завершение цикла можно, поскольку есть параметр, принимающий значения в фундированном множестве и убывающий на каждом шаге цикла.

Вот пример олимпиадной задачи, где по существу такое рассуждение и используется.

Бизнесмен заключил с чёртом сделку: каждый день он даёт чёрту одну монету, и в обмен получает любой набор монет по своему выбору, но все эти монеты меньшего достоинства (видов монет конечное число). Менять (или получать) деньги в другом месте бизнесмен не может. Когда монет больше не останется, бизнесмен проигрывает. Докажите, что рано или поздно чёрт выиграет, каков бы ни был начальный набор монет у бизнесмена.

Решение: пусть имеется k видов монет. Искомый параметр определим так: посчитаем, сколько монет каждого вида есть у бизнесмена (n_1 — число монет минимального достоинства, n_2 — число следующих, и так далее до n_k). Заметим, что в результате встречи с чёртом набор $\langle n_1, \ldots, n_k\rangle$ уменьшается (в смысле введённого нами порядка, когда мы сравниваем сначала последние члены, затем предпоследние и т. д.). Поскольку множество $\mathbb{N}^k$ фундировано, этот процесс должен оборваться.

Задача 86. *Имеется конечная последовательность нулей и единиц. За один шаг разрешается сделать такое действие: найти в ней группу* 01 *и заменить на* 100...00 *(при этом можно написать сколько угодно нулей). Докажите, что такие шаги нельзя выполнять бесконечно много раз.*

Задача 87. *Рассмотрим множество всех слов русского алфавита (а точнее, всех конечных последовательностей русских букв, независимо от смысла) с лексикографическим порядком (см. с. 38). Будет ли это множество фундировано?*

Задача 88. *Рассмотрим множество тех невозрастающих последовательностей натуральных чисел, в которых все члены, начиная с некоторого, равны нулю. Введём в нём порядок так: сначала сравниваем первые члены, при равенстве первых вторые и т. д. Докажите, что это (линейно) упорядоченное множество фундировано.*

Задача 89. *Рассмотрим множество всех многочленов от одной переменной x, коэффициенты которых — натуральные числа. Упорядочим его так: многочлен P больше многочлена Q, если $P(x) > Q(x)$ для всех достаточно больших x. Покажите, что это определение задаёт линейный*

порядок и что получающееся упорядоченное множество фундировано.

Фундированные линейно упорядоченные множества называются *вполне упорядоченными*, а соответствующие порядки — *полными*. Для линейных порядков понятия наименьшего и минимального элемента совпадают, так что во вполне упорядоченном множестве всякое непустое подмножество имеет наименьший элемент.

Заметим, что частично упорядоченное множество, в котором всякое непустое подмножество имеет наименьший элемент, автоматически является линейно упорядоченным (в самом деле, всякое двухэлементное множество имеет наименьший элемент, поэтому любые два элемента сравнимы).

Примеры вполне упорядоченных множеств: $\mathbb{N}$, $\mathbb{N}+k$ (здесь k обозначает конечное линейно упорядоченное множество из k элементов), $\mathbb{N}+\mathbb{N}$, $\mathbb{N}\times\mathbb{N}$.

Наша цель — понять, как могут быть устроены вполне упорядоченные множества. Начнём с нескольких простых замечаний.

- Вполне упорядоченное множество имеет наименьший элемент. (Непосредственное следствие определения.)
- Для каждого элемента x вполне упорядоченного множества (кроме наибольшего) есть непосредственно следующий за ним элемент y (это значит, что $y > x$, но не существует z, для которого $y > z > x$). В самом деле, если множество всех элементов, больших x, непусто, то в нём есть минимальный элемент y, который и будет искомым. Такой элемент логично обозначать $x+1$, следующий за ним — $x+2$ и т. д.
- Некоторые элементы вполне упорядоченного множества могут не иметь непосредственно предыдущего. Например, в множестве $\mathbb{N}+\mathbb{N}$ есть два элемента, не имеющих непосредственно предыдущего (наименьший элемент, а также наименьший элемент второй копии натурального ряда). Такие элементы называют *предельными*.
- Всякий элемент упорядоченного множества имеет вид $z+n$, где z — предельный, а n — натуральное число (обозначение $z+n$ понимается в описанном выше смысле). В самом деле, если z не предельный, возьмём предыдущий, если и он непредельный — то его предыдущий и т. д., пока не дойдём до предельного (бесконечно продолжаться это не может, так как множество вполне упорядочено). Очевидно, такое представление однозначно (у элемента может быть только один непосредственно предыдущий).
- Любое ограниченное сверху множество элементов вполне упорядоченного множества имеет точную верхнюю грань. (Как обычно, подмножество X частично упорядоченного множества A называется *ограниченным сверху*, если оно имеет *верхнюю границу*, т. е. элемент $a \in A$,

для которого $x \leqslant a$ при всех $x \in X$. Если среди всех верхних границ данного подмножества есть наименьшая, то она называется *точной верхней гранью*.)

В самом деле, множество всех верхних границ непусто и потому имеет наименьший элемент. (Заметим в скобках, что вопрос о точной нижней грани для вполне упорядоченного множества тривиален, так как всякое множество имеет наименьший элемент.)

Пусть A — произвольное вполне упорядоченное множество. Его наименьший элемент обозначим через 0. Следующий за ним элемент обозначим через 1, следующий за 1 — через 2 и т. д. Если множество конечно, процесс этот оборвётся. Если бесконечно, посмотрим, исчерпали ли мы все элементы множества A. Если нет, возьмём минимальный элемент из оставшихся. Обозначим его ω. Следующий за ним элемент (если он есть) обозначим $\omega + 1$, затем $\omega+2$ и т. д. Если и на этом множество не исчерпается, то возьмём наименьший элемент из оставшихся, назовём его $\omega \cdot 2$, и повторим всю процедуру. Затем будут $\omega \cdot 3$, $\omega \cdot 4$ и т. д. Если и на этом множество не кончится, минимальный из оставшихся элементов назовём ω^2. Затем пойдут $\omega^2 + 1$, $\omega^2 + 2$, …, $\omega^2 + \omega$, …, $\omega^2 + \omega \cdot 2$, …, $\omega^2 \cdot 2$, …, $\omega^2 \cdot 3$, …, ω^3, … (мы не поясняем сейчас подробно обозначения).

Что, собственно говоря, доказывает это рассуждение? Попытаемся выделить некоторые утверждения. При этом полезно такое определение: если линейно упорядоченное множество A разбито на две (непересекающиеся) части B и C, причём любой элемент B меньше любого элемента C, то B называют *начальным отрезком* множества A. Другими словами, подмножество B линейно упорядоченного множества A является начальным отрезком, если любой элемент B меньше любого элемента $A \setminus B$. Ещё одна переформулировка: $B \subset A$ является начальным отрезком, если из $a, b \in A$, $b \in B$ и $a \leqslant b$ следует $a \in B$. Заметим, что начальный отрезок может быть пустым или совпадать со всем множеством.

Отметим сразу же несколько простых свойств начальных отрезков:

- Начальный отрезок вполне упорядоченного множества (как, впрочем, и любое подмножество) является вполне упорядоченным множеством.
- Начальный отрезок начального отрезка есть начальный отрезок исходного множества.
- Объединение любого семейства начальных отрезков (в одном и том же упорядоченном множестве) есть начальный отрезок того же множества.
- Если x — произвольный элемент вполне упорядоченного множества A, то множества $[0, x)$ (все элементы множества A, меньшие x)

и $[0, x]$ (элементы множества A, меньшие или равные x) являются начальными отрезками.

- Всякий начальный отрезок I вполне упорядоченного множества A, не совпадающий со всем множеством, имеет вид $[0, x)$ для некоторого $x \in A$. (В самом деле, если $I \neq A$, возьмём наименьший элемент x в множестве $A \setminus I$. Тогда все меньшие элементы принадлежат I, сам x не принадлежит I и все бóльшие x элементы не принадлежат I, иначе получилось бы противоречие с определением начального отрезка.)
- Любые два начальных отрезка вполне упорядоченного множества сравнимы по включению, т. е. один есть подмножество другого. (Следует из предыдущего.)
- Начальные отрезки вполне упорядоченного множества A, упорядоченные по включению, образуют вполне упорядоченное множество. Это множество состоит из наибольшего элемента (всё A) и остальной части, изоморфной множеству A. (В самом деле, начальные отрезки множества A, не совпадающие с A, имеют вид $[0, x)$, и соответствие $[0, x) \leftrightarrow x$ будет изоморфизмом.)

Возвратимся к нашему рассуждению с последовательным выделением различных элементов из вполне упорядоченного множества. Его первую часть можно считать доказательством такого утверждения: если вполне упорядоченное множество бесконечно, то оно имеет начальный отрезок, изоморфный ω. (Говоря о множестве натуральных чисел вместе с порядком, обычно употребляют обозначение ω, а не $\mathbb{N}$.)

Но на этом наше рассуждение не оканчивается. Его следующая часть может считаться доказательством такого факта: либо A изоморфно некоторому начальному отрезку множества ω^2, либо оно имеет начальный отрезок, изоморфный ω^2. (Здесь ω^2 — вполне упорядоченное множество пар натуральных чисел: сравниваются сначала вторые компоненты пар, а при их равенстве — первые.)

Вообще верно такое утверждение: для любых двух вполне упорядоченных множеств одно изоморфно начальному отрезку другого, и доказательство состоит более или менее в повторении проведённого рассуждения. Но чтобы сделать это аккуратно, нужна некоторая подготовка.

ГЛАВА III

Логика высказываний

§1. Высказывания и операции

"Если число π рационально, то π — алгебраическое число. Но оно не алгебраическое. Значит, π не рационально." Мы не обязаны знать, что такое число π, какие числа называют рациональными и какие алгебраическим, чтобы признать, что это рассуждение правильно — в том смысле, что из двух сформулированных посылок действительно вытекает заключение. Такого рода ситуации — когда некоторое утверждение верно независимо от смысла входящий в него высказываний — составляют предмет *логики высказываний*.

Наши рассмотрения будут иметь вполне точный математический характер, хотя мы начнём с неформальных мотивировок.

Высказывания могут быть *истинными* и *ложными*. Например, "$2^{16}+1$ — простое число" — истинное высказывание, а "$2^{32}+1$ — простое число" — ложное (это число делится на 641). Про высказывание "существует бесконечно много простых p, для которых $p+2$ — также простое" никто не берётся сказать наверняка, истинно оно или ложно. Заметим, что "x делится на 2" в этом смысле не является высказыванием, пока не сказано, чему равно x; при разных x получаются разные высказывания, одни истинные (при чётном x), другие — ложные (при нечётном x).

Высказывания можно соединять друг с другом с помощью "логических связок". Эти связки имеют довольно странные, но традиционные названия и обозначения (табл. 1). Отметим также, что в импликации $A \Rightarrow B$ высказывание A называют *посылкой*, или *антецедентом импликации*, а B — *заключением*, или *консеквентом*.

Говорят также, что высказывание имеет *истинностное значение* **И** (истина), если оно истинно, или **Л** (ложь), если оно ложно. Иногда вместо **И** употребляется буква **T** (true) или число 1, а вместо **Л** — буква **F** (false) или число 0. (С первого взгляда идея произвольным образом выбрать числа 0 и 1 кажется дикой — какая бы польза могла быть от, скажем, сложения истинностных значений? Удивительным образом в последние годы обнаружилось, что такая польза есть, и если оперировать с истиной и ложью как элементами конечного поля, можно получить много неожиданных результатов. Но это выходит за рамки нашей книги.)

связка	обозначение	название
A и B	$A\&B$ $A \wedge B$ A `and` B	конъюнкция
A или B	$A \vee B$ A `or` B	дизъюнкция
не A A неверно	$\neg A$ $\sim A$ $\overline{A}$ `not` A	отрицание
из A следует B если A, то B A влечёт B B — следствие A	$A \to B$ $A \Rightarrow B$ $A \supset B$ `if` A `then` B	импликация следование

Приложение 1. Логические связки, обозначения и названия.

Логические связки позволяют составлять сложные высказывания из простых. При этом истинность составного высказывания определяется истинностью его частей в соответствии с таблицей 2.

A	B	$A \wedge B$	$A \vee B$	$A \to B$
Л	**Л**	**Л**	**Л**	**И**
Л	**И**	**Л**	**И**	**И**
И	**Л**	**Л**	**И**	**Л**
И	**И**	**И**	**И**	**И**

A	$\neg A$
Л	**И**
И	**Л**

Приложение 2. Таблицы истинности для логических связок.

Те же правила можно изложить словесно. Высказывание $A \wedge B$ истинно, если оба высказывания A и B истинны. Высказывание $A \vee B$ истинно, если хотя бы одно из высказываний A и B истинно. Высказывание $A \to B$ ложно в единственном случае: если A истинно, а B ложно. Наконец, $\neg A$ истинно в том и только том случае, когда A ложно.

Из всех связок больше всего вопросов вызывает импликация. В самом деле, не очень понятно, почему надо считать, скажем, высказывания "если $2 \times 2 = 5$, то $2 \times 2 = 4$" и "если $2 \times 2 = 5$, то $3 \times 3 = 1$" истинными. (Именно так говорят наши таблицы: **Л** $\to$ **И** = **Л** $\to$ **Л** = **И**.) Следующий пример показывает, что в таком определении есть смысл.

Общепризнано, что если число x делится на 4, то оно делится на 2. Это

означает, что высказывание

$$(x \text{ делится на } 4) \ \rightarrow \ (x \text{ делится на } 2)$$

истинно при всех x. Подставим сюда $x = 5$: обе части ложны, а утверждение в целом истинно. При $x = 6$ посылка импликации ложна, а заключение истинно, и вся импликация истинна. Наконец, при $x = 8$ посылка и заключение истинны и импликация в целом истинна. С другой стороны, обратное утверждение (если x делится на 2, то x делится на 4) неверно, и число 2 является контрпримером. При этом посылка импликации истинна, заключение ложно, и сама импликация ложна. Таким образом, если считать, что истинность импликации определяется истинностью её частей (а не наличием между ними каких-то причинно-следственных связей), то все строки таблицы истинности обоснованы. Чтобы подчеркнуть такое узко-формальное понимание импликации, философски настроенные логики называют её "материальной импликацией".

Теперь от неформальных разговоров перейдём к определениям. Элементарные высказывания (из которых составляются более сложные) мы будем обозначать маленькими латинскими буквами и называть *пропозициональными переменными*. Из них строятся *пропозициональные формулы* по таким правилам:

- Всякая пропозициональная переменная есть формула.
- Если A — пропозициональная формула, то $\neg A$ — пропозициональная формула.
- Если A и B — пропозициональные формулы, то $(A \wedge B)$, $(A \vee B)$ и $(A \rightarrow B)$ — пропозициональные формулы.

Можно ещё сказать так: формулы образуют минимальное множество, обладающее указанными свойствами (слово "минимальное" здесь существенно: ведь если бы мы объявили любую последовательность переменных, скобок и связок формулой, то эти три свойства были бы тоже выполнены).

Пусть формула φ содержит n пропозициональных переменных $p_1, p_2, \ldots, p_n$. Если подставить вместо этих переменных истинностные значения (**И** или **Л**), то по таблицам можно вычислить истинностное значение формулы в целом. Таким образом, формула задаёт некоторую функцию от n аргументов, каждый из которых может принимать значения **Л** и **И**. Значения функции также лежат в множестве $\{\mathbf{Л}, \mathbf{И}\}$, которое мы будем обозначать $\mathbb{B}$. Мы будем следовать уже упоминавшейся традиции и отождествлять **И** с единицей, а **Л** — с нулём, тем самым $\mathbb{B}$ есть $\{0, 1\}$. Формула φ задаёт отображение типа $\mathbb{B}^n \rightarrow \mathbb{B}$. Такие отображения называют также *булевыми функциями n аргументов*.

Пример. Рассмотрим формулу $(p \wedge (q \wedge \neg r))$. Она истинна в единственном случае — когда p и q истинны, а r ложно (см. таблицу 3).

p	q	r	$\neg r$	$(q \wedge \neg r)$	$(p \wedge (q \wedge \neg r))$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0

Приложение 3. Таблица истинности для $(p \wedge (q \wedge \neg r))$.

Некоторые формулы выражают логические законы — составные высказывания, истинные независимо от смысла их частей. Такие формулы (истинные при всех значениях входящих в них переменных) называют *тавтологиями*.

Пример. Формула $((p \wedge q) \to p)$ является тавтологией (это можно проверить, например, составив таблицу). Она выражает такой логический закон: из конъюнкции утверждений следует первое из них.

Задача 90. *Как выглядит симметричное утверждение для дизъюнкции и какая формула его выражает?*

Две формулы называют *эквивалентными*, если они истинны при одних и тех же значениях переменных (другими словами, если они задают одну и ту же булеву функцию). Например, формула $(p \wedge (p \to q))$ истинна лишь при $p = q =$ **И**, и потому эквивалентна формуле $(p \wedge q)$.

Рассмотрим формулу $((p \wedge q) \vee q)$. Она истинна, если переменная q истинна, и ложна, если переменная q ложна. Хотелось бы сказать, что она эквивалентна формуле q, но тут есть формальная трудность: она содержит две переменные и потому задаёт функцию от двух аргументов (типа $\mathbb{B} \times \mathbb{B} \to \mathbb{B}$), в то время как формула q задаёт функцию одного аргумента. Мы не будем обращать на это внимания и будем считать эти формулы эквивалентными. Вообще, если есть список переменных $p_1, \ldots, p_n$, содержащий все переменные некоторой формулы φ (и, возможно, ещё какие-то переменные), можно считать, что формула φ задаёт функцию от n аргументов, возможно, на деле зависящую не от всех аргументов (постоянную по некоторым аргументам)

После сделанных оговорок легко проверить следующий факт: формулы φ и ψ эквивалентны тогда и только тогда, когда формула $((\varphi \to \psi) \wedge (\psi \to \varphi))$ является тавтологией. Используя сокращение $(p \leftrightarrow q)$ для $((p \to q) \wedge (q \to p))$, можно записывать утверждения об эквивалентности формул в виде тавтологий. Вот несколько таких эквивалентностей:

Теорема 16. *Формулы*

$$
\begin{aligned}
(p \wedge q) &\leftrightarrow (q \wedge p);\\
((p \wedge q) \wedge r) &\leftrightarrow (p \wedge (q \wedge r));\\
(p \vee q) &\leftrightarrow (q \vee p);\\
((p \vee q) \vee r) &\leftrightarrow (p \vee (q \vee r));\\
(p \wedge (q \vee r)) &\leftrightarrow ((p \wedge q) \vee (p \wedge r));\\
(p \vee (q \wedge r)) &\leftrightarrow ((p \vee q) \wedge (p \vee r));\\
\neg(p \wedge q) &\leftrightarrow (\neg p \vee \neg q);\\
\neg(p \vee q) &\leftrightarrow (\neg p \wedge \neg q);\\
(p \vee (p \wedge q)) &\leftrightarrow p;\\
(p \wedge (p \vee q)) &\leftrightarrow p;\\
(p \to q) &\leftrightarrow (\neg q \to \neg p);\\
p &\leftrightarrow \neg\neg p
\end{aligned}
$$

являются тавтологиями.

Доказательство. Первые четыре эквивалентности выражают коммутативность и ассоциативность конъюнкции и дизъюнкции. Проверим, например, вторую: левая и правая части истинны в единственном случае (когда все переменные истинны), и потому эквивалентны. (Для дизъюнкции удобнее смотреть, когда она ложна.)

Две следующие эквивалентности утверждают дистрибутивность — заметим, что в отличие от сложения и умножения в кольцах здесь верны оба свойства дистрибутивности. Проверить эквивалентность легко, если отдельно рассмотреть случаи истинного и ложного p.

Следующие два свойства, *законы Де Моргана*, легко проверить, зная, что конъюнкция истинна, а дизъюнкция ложна лишь в одном случае. Эти свойства иногда выражают словами: "конъюнкция двойственна дизъюнкции".

Далее следуют два очевидных *закона поглощения* (один из них мы уже упоминали).

За ними идёт правило *контрапозиции*, которое говорит, в частности, что утверждения "если x совершенно, то x чётно" и "если x нечётно, то x

несовершенно" равносильны. Хотя оно и очевидно проверяется с помощью таблиц истинности, с ним связаны любопытные парадоксы. Вот один из них.

Биолог А выдвинул гипотезу: все вороны чёрные. Проверяя её, он вышел во двор и обнаружил на дереве ворону. Она оказалось чёрной. Биолог А радуется — гипотеза подтверждается. Его друг математик Б переформулировал гипотезу так: все не-чёрные предметы — не вороны (применив наше правило контрапозиции) и не стал выходить во двор, а открыл холодильник и нашёл там оранжевый предмет. Он оказался апельсином, а не вороной. Математик Б обрадовался — гипотеза подтверждается — и позвонил биологу А. Тот удивляется — у него тоже есть апельсин в холодильнике, но с его точки зрения никакого отношения к его гипотезе апельсин не имеет . . .

Другой парадокс: с точки зрения формальной логики утверждения "кто не с нами, тот против нас" и "кто не против нас, тот с нами" равносильны.

Последнее (и очевидное) правило $p \leftrightarrow \neg\neg p$ называется *снятием двойного отрицания.* □

Задача 91. *Перечисленные эквивалентности соответствуют равенствам для множеств: например, первая гарантирует, что* $P \cap Q = Q \cap P$ *для любых множеств* P *и* Q*. Какие утверждения соответствуют остальным эквивалентностям?*

Задача 92. *Две формулы, содержащие только переменные и связки* $\wedge$, $\vee$ *и* $\neg$*, эквивалентны. Докажите, что они останутся эквивалентными, если всюду заменить* $\wedge$ *на* $\vee$ *и наоборот.*

Далеко не все тавтологии имеют ясный интуитивный смысл. Например, формула $(p \to q) \vee (q \to p)$ является тавтологией (если одно из утверждений p и q ложно, то из него следует всё, что угодно; если оба истинны, то тем более формула истинна), хотя и отчасти противоречит нашей интуиции — почему, собственно, из двух никак не связанных утверждений одно влечёт другое? Ещё более загадочна тавтология

$$((p \to q) \to p) \to p$$

(хотя её ничего не стоит проверить с помощью таблиц истинности).

Отступление о пользе скобок. На самом деле наше определение истинности содержит серьёзный пробел. Чтобы обнаружить его, зададим себе вопрос: зачем нужны скобки в формулах? Представим себе, что мы изменим определение формулы, и будем говорить, что $P \wedge Q$ и $P \vee Q$ являются формулами для любых P и Q. Останутся ли наши рассуждения в силе?

Легко понять, что мы столкнёмся с трудностью при определении булевой функции, соответствующей формуле. В этом определении мы подставляли нули и единицы на место переменных и затем вычисляли значение формулы с помощью таблиц истинности для связок. Но теперь, когда мы изменили определение формулы, формула $p \wedge q \vee r$ может быть получена двумя способами — из формул $p \wedge q$ и r с помощью операции $\vee$ и из формул p и $q \vee r$ с помощью операции $\wedge$. Эти два толкования дадут разный результат при попытке вычислить значение $0 \wedge 0 \vee 1$.

Из сказанного ясно, что скобки нужны, чтобы гарантировать однозначность синтаксического разбора формулы. Точнее говоря, верно такое утверждение:

Теорема 17 (однозначность разбора). *Пропозициональная формула, не являющаяся переменной, может быть представлена ровно в одном из четырёх видов* $(A \wedge B)$, $(A \vee B)$, $(A \to B)$ *или* $\neg A$, *где* A *и* B *— некоторые формулы, причём* A *и* B *(в первых трёх случаях) восстанавливаются однозначно.*

Доказательство. Формальное доказательство можно провести так: назовём *скобочным итогом* разницу между числом открывающихся и закрывающихся скобок. Индукцией по построению формулы легко доказать такую лемму:

Лемма. Скобочный итог формулы равен нулю. Скобочный итог любого начала формулы неотрицателен и равен нулю, лишь если это начало совпадает со всей формулой, пусто или состоит из одних символов отрицания.

Слова "индукцией по построению" означают, что мы проверяем утверждение для переменных, а также доказываем, что если оно верно для формул A и B, то оно верно и для формул $(A \wedge B)$, $(A \vee B)$, $(A \to B)$ и $\neg A$.

После того как лемма доказана, разбор формулы проводится так: если она начинается с отрицания, то может быть образована лишь по четвертому правилу. Если же она начинается со скобки, то надо скобку удалить, а потом искать непустое начало, имеющее нулевой скобочный итог и не оканчивающееся на знак логической операции. Такое начало единственно (как легко проверить, используя лемму). Это начало и будет первой частью формулы. Тем самым формула разбирается однозначно. □

Нет смысла вдаваться в подробности этого (несложного) рассуждения: вообще-то алгоритмы разбора формул — это отдельная большая и практически важная тема (в первую очередь в связи с компиляторами). Приведённый нами алгоритм далеко не оптимален. С другой стороны, мы вообще можем обойти эту проблему, потребовав, чтобы при записи формул левая и

правая скобки, окружающие формулу, связывались линией — тогда однозначность разбора формулы не вызывает вопросов, и больше ничего нам не надо.

В дальнейшем мы будем опускать скобки, если они либо не играют роли (например, можно написать конъюнкцию трёх членов, не указывая порядок действий в силу ассоциативности), либо ясны из контекста.

Задача 93. *Польский логик Лукасевич предлагал обходиться без скобок, записывая в формулах сначала знак операции, а потом операнды (без пробелов и разделителей). Например,* $(a+b)\times(c+(d\times e))$ *в его обозначениях запишется как* $\times+ab+c\times de$. *Эту запись ещё называют* польской *записью. Обратная польская запись отличается от неё тем, что знак операции идёт после операндов. Покажите, что в обоих случаях порядок действий восстанавливается однозначно.*

§2. Полные системы связок

Рассматриваемая нами система пропозициональных связок ($\wedge$, $\vee$, $\to$, $\neg$) *полна* в следующем смысле:

Теорема 18 (Полнота системы связок). *Любая булева функция n аргументов может быть записана в виде пропозициональной формулы.*

Доказательство. Проще всего пояснить это на примере. Пусть, например, булева функция $\varphi(p,q,r)$ задана таблицей 4.

p	q	r	$\varphi(p,q,r)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$(\neg p\wedge\neg q\wedge\neg r)\vee$$
$$\vee(\neg p\wedge q\wedge r)\vee$$
$$\vee(p\wedge q\wedge r)$$

Приложение 4. Булева функция и задающая её формула.

В таблице есть три строки с единицами в правой колонке — три случая, когда булева функция истинна (равна 1). Напишем три конъюнкции, каждая из которых покрывает один случай (а в остальных строках ложна), и соединим их дизъюнкцией. Нужная формула построена.

Ясно, что аналогичная конструкция применима для любой таблицы (с любым числом переменных). □

Для формул подобного вида есть специальное название: формулы в *дизъюнктивной нормальной форме*. Более подробно: *литералом* называется переменная или отрицание переменной, *конъюнктом* называется произвольная конъюнкция литералов, а дизъюнктивной нормальной формой называется дизъюнкция конъюнктов. В нашем случае в каждый конъюнкт входит n литералов (где n — число переменных), а число конъюнктов равно числу строк с единицами и может меняться от нуля (тогда, правда, получается не совсем формула, а "пустая дизъюнкция", и её можно заменить какой-нибудь всегда ложной формулой типа $p \wedge \neg p$) до 2^n (если булева функция всегда истинна).

Задача 94. *Длина построенной в доказательстве теоремы 18 формулы зависит от числа единиц: формула будет короткой, если единиц в таблице мало. А как написать (сравнительно) короткую формулу, если в таблице мало нулей, а в основном единицы?*

Иногда полезна *конъюнктивная нормальная форма*, которая представляет собой конъюнкцию *дизъюнктов*. Каждый дизъюнкт состоит из литералов, соединённых дизъюнкциями. Теорему 18 можно теперь усилить так:

Теорема 19. *Всякая булева функция может быть выражена формулой, находящейся в дизъюнктивной нормальной форме, а также формулой, находящейся в конъюнктивной нормальной форме.*

Доказательство. Первая часть утверждения уже доказана. Вторая часть аналогична первой, надо только для каждой строки с нулём написать подходящий дизъюнкт.

Можно также представить функцию $\neg\varphi$ в дизъюнктивной нормальной форме, а затем воспользоваться законами Де Моргана, чтобы внести отрицание внутрь. □

Задача 95. *Проведите второй вариант рассуждения подробно.*

Вообще говоря, определение нормальной формы не требует, чтобы в каждом конъюнкте (или дизъюнкте) встречались все переменные. (Повторять переменную больше одного раза смысла нет; если, например, переменная и её отрицание входят в одну конъюнкцию, то эта конъюнкция всегда ложна и её можно выбросить.)

Задача 96. *Приведите пример булевой функции n аргументов, у которой любая дизъюнктивная или конъюнктивная нормальная форма содержит лишь члены длины n. (Указание: рассмотрите функцию, которая меняет своё значение при изменении значения любой переменной.)*

Заметим, что при доказательстве теоремы 18 мы обошлись без импликации. Это и не удивительно, так как она выражается через дизъюнкцию и отрицание:

$$(p \to q) \ \leftrightarrow \ (\neg p \vee q)$$

(проверьте!). Мы могли бы обойтись только конъюнкцией и отрицанием, так как

$$(p \vee q) \ \leftrightarrow \ \neg(\neg p \wedge \neg q),$$

или только дизъюнкцией и отрицанием, так как

$$(p \wedge q) \ \leftrightarrow \ \neg(\neg p \vee \neg q)$$

(обе эквивалентности вытекают из законов Де Моргана; их легко проверить и непосредственно). Как говорят, система связок $\wedge, \neg$, а также система связок $\vee, \neg$ являются *полными*. (По определению это означает, что с их помощью можно записать любую булеву функцию.)

Задача 97. *Докажите, что система связок $\neg, \to$ полна. (Указание: как записать через них дизъюнкцию?)*

А вот без отрицания обойтись нельзя. Система связок $\wedge, \vee, \to$ неполна — и по очень простой причине: если все переменные истинны, то любая их комбинация, содержащая только указанные связки, истинна. (Как говорят, все эти связки "сохраняют единицу".)

Задача 98. *Легко понять, что любая формула, составленная только с помощью связок $\wedge$ и $\vee$, задаёт монотонную булеву функцию (в том смысле, что от увеличения значения любого из аргументов значение функции может только возрасти — или остаться прежним). Покажите, что любая монотонная булева функция может быть выражена формулой, содержащей только $\wedge$ и $\vee$.*

Задача 99. *Пусть $\varphi \to \psi$ — тавтология. Покажите, что найдётся формула τ, которая включает в себя только общие для φ и ψ переменные, для которой формулы $(\varphi \to \tau)$ и $(\tau \to \psi)$ являются тавтологиями. (Более общий вариант этого утверждения, в котором рассматриваются формулы с кванторами, называется* леммой Крейга.*)*

В принципе мы не обязаны ограничиваться четырьмя рассмотренными связками. Любая булева функция может играть роль связки. Например, можно рассмотреть связку (p `notand` q), задаваемую эквивалентностью

$$(p \texttt{ notand } q) \leftrightarrow \neg(p \wedge q)$$

(словами: (p `notand` q) ложно, лишь если p и q истинны). Через неё выражается отрицание (p `notand` p), после чего можно выразить конъюнкцию, а затем, как мы знаем, и вообще любую функцию. (Знакомые с цифровыми логическими схемами малого уровня интеграции хорошо знакомы с этим утверждением: достаточно большой запас схем И-НЕ позволяет реализовать любую требуемую зависимость выхода от входов.)

Другая интересная полная система связок — сложение по модулю 2, конъюнкция и константа 1 (которую можно считать 0-арной связкой, задающей функцию от нуля аргументов). Представленные в этой системе булевы функции становятся полиномами с коэффициентами в кольце вычетов по модулю 2. Идея рассматривать булевы функции как полиномы (оказавшаяся неожиданно плодотворной в последние годы) была высказана в 1927 г. российским математиком Иваном Ивановичем Жегалкиным.

Назовём *мономом* конъюнкцию любого набора переменных или константу 1 (которую естественно рассматривать как конъюнкцию нуля переменных). Название это естественно, так как при наших соглашениях (1 обозначает истину, 0 — ложь) конъюнкция соответствует умножению.

Назовём *полиномом* сумму таких мономов по модулю 2 (это значит, что $0 \oplus 0 = 0$, $0 \oplus 1 = 1 \oplus 0 = 1$ и $1 \oplus 1 = 0$). Ясно, что два повторяющихся монома можно сократить (ведь сложение по модулю 2), так что будем рассматривать только полиномы без повторяющихся мономов. При этом, естественно, порядок членов в мономе (как и порядок мономов в полиноме) роли не играет, их можно переставлять.

Теорема 20 (о полиномах Жегалкина). *Всякая булева функция однозначно представляется таким полиномом.*

Доказательство. Существование искомого полинома следует из теоремы 19, так как конъюнкция есть умножение, отрицание — прибавление единицы, а дизъюнкцию можно через них выразить (получится $p + q + pq$). Надо только заметить, что степени не нужны: переменные принимают значения 0 и 1, так что x^n можно заменить на x.

Можно также сослаться на известное из алгебры утверждение о том, что всякая функция с аргументами из конечного поля (в данном случае это двухэлементное поле вычетов по модулю 2) задаётся полиномом. (Отсюда, кстати, получается новое доказательство теоремы 18.)

Далее можно заметить, что полиномов столько же, сколько булевых функций, а именно 2^{2^n}. В самом деле, булева функция может принимать любое из двух значений в каждой из 2^n точек булева куба $\mathbb{B}^n$, а многочлен может включать или не включать любой из 2^n мономов. (Мономов ровно 2^n, потому что каждый моном включает или не включает любую из n переменных.) Поэтому избытка полиномов нет, и если любая функция представима полиномом, то единственным образом.

Можно и не ссылаться на сведения из алгебры и теорему 19, а дать явную конструкцию. Это удобно сделать индукцией по n. Пусть мы уже умеем представлять любую булеву функцию от $n-1$ аргументов с помощью полинома. Тогда $\varphi(p_1, \ldots, p_n)$ можно представить как

$$\varphi(p_1, \ldots, p_n) = \varphi(0, p_2, \ldots, p_n) + [\varphi(0, p_2, \ldots, p_n) + \varphi(1, p_2, \ldots, p_n)]p_1$$

(проверьте). Остаётся заметить, что правую часть можно представить полиномом по предположению индукции.

Для единственности также есть другое доказательство: пусть два многочлена (имеющие степень 1 по каждой переменной) равны при всех значениях переменных. Тогда их сумма (или разность — вычисления происходят по модулю 2) является ненулевым многочленом (содержит какие-то мономы), но тождественно равна нулю. Так не бывает, и это легко доказать по индукции. В самом деле, любой многочлен $A(p_1, \ldots, p_n)$ можно представить в виде

$$A(p_1, \ldots, p_n) = B(p_2, \ldots, p_n) + p_1 C(p_2, \ldots, p_n),$$

где B и C — многочлены от меньшего числа переменных. Подставляя сначала $p_1 = 0$, а затем $p_1 = 1$, убеждаемся, что многочлены B и C равны нулю во всех точках, и потому (согласно предположению индукции) равны нулю как многочлены (не содержат мономов). □

Задача 100. *Пусть F — произвольное поле. Назовём* мультилинейной *функцией полином от n переменных с коэффициентами из F, в котором все показатели степеней равны либо 0, либо 1. (Таким образом, каждый моном в ней есть произведение коэффициента и некоторого набора переменных без повторений.) Будем рассматривать $\mathbb{B} = \{0, 1\}$ как подмножество F. Докажите, что всякая булева функция $\mathbb{B}^n \to \mathbb{B}$ однозначно продолжается до мультилинейной функции $F^n \to F$, и коэффициенты мультилинейной функции можно считать целыми числами.*

Если рассматривать произвольные булевы функции в качестве связок, возникает вопрос: в каком случае набор булевых функций образует полный базис? (Это значит, что любая булева функция представляется в виде

композиции функций из набора, т. е. записывается в виде формулы, где связками служат функции набора.) Подобные вопросы вызывали в своё время большой интерес и были хорошо изучены. Начальным этапом явилось такое утверждение:

Теорема 21 (критерий Поста). *Набор булевых функций является полным тогда и только тогда, когда он не содержится целиком ни в одном из пяти следующих "предполных классов":*

- *монотонные функции;*
- *функции, сохраняющие нуль;*
- *функции, сохраняющие единицу;*
- *линейные функции;*
- *самодвойственные функции.*

(Функция f монотонна*, если она монотонно неубывает по каждому из своих аргументов. Функция f* сохраняет нуль/единицу*, если $f(0,\dots,0)=0$ (соответственно $f(1,\dots,1)=1$). Функция f* линейна*, если она представима многочленом, в котором все мономы содержат не более одной переменной. Наконец, функция f называется* самодвойственной*, если $f(1-p_1,\dots,1-p_n)=1-f(p_1,\dots,p_n)$.)*

Доказательство. Если набор содержится в одном из классов, то и все композиции также не выходят за пределы этого класса (легко проверить для каждого из классов в отдельности) и поэтому набор не является полным. Докажем обратное утверждение. Пусть для каждого класса выбрана какая-то функция, в нём не лежащая. Убедимся, что с помощью комбинаций выбранных функций можно получить все булевы функции.

У нас есть функция, не сохраняющая нуль. Подставим вместо всех аргументов одну и ту же переменную. Получится функция от одного аргумента, отображающая нуль в единицу, то есть либо константа 1, либо отрицание. Сделав то же самое с функцией, не сохраняющей единицу, получим либо константу нуль, либо отрицание. Таким образом, у нас либо есть отрицание, либо обе константы 0 и 1.

Если есть обе константы, то всё равно можно получить отрицание. Возьмём немонотонную функцию. Легко понять, что она должна менять значение с единицы на нуль при изменении какого-то одного аргумента с нуля на единицу (в самом деле, будем увеличивать аргументы по одному, в какой-то момент значение функции уменьшится.) Зафиксировав значения остальных аргументов (ведь мы считаем, что константы есть), получаем отрицание.

Имея отрицание и несамодвойственную функцию, легко получить константы (если их не было). В самом деле, несамодвойственность означает,

что $f(x_1,\dots,x_n)=f(1-x_1,\dots,1-x_n)$ для каких-то значений $x_1,\dots,x_n\in\{0,1\}$. Вместо нулевых значений переменных $x_1,\dots,x_n$ подставим p, вместо единиц подставим $\neg p$, получится одна из констант. Вторая получится отрицанием.

Теперь у нас есть константы, отрицание и нелинейная функция $f(p_1,\dots,p_n)$. Нелинейность означает, что в её представлении в виде многочлена есть моном, состоящий более чем из одной переменной. Пусть, например, этот моном содержит переменные p_1 и p_2. Сгруппируем члены по четырём группам и получим выражение

$$p_1p_2A(p_3,\dots)+p_1B(p_3,\dots)+p_2C(p_3,\dots)+D(p_3,\dots).$$

При этом многочлен $A(p_3,\dots)$ заведомо отличен от нуля, поэтому можно так подставить константы вместо $p_3,\dots,p_n$, чтобы первое слагаемое не обратилось в нуль. Тогда получим либо p_1p_2+d, либо $p_1p_2+p_1+d$, либо $p_1p_2+p_2+d$, либо $p_1p_2+p_1+p_2+d$. Свободный член d можно менять, если нужно (у нас есть отрицание), так что получается либо p_1p_2 (конъюнкция, и всё доказано), либо $p_1p_2+p_1=p_1(p_2+1)=p_1\wedge\neg p_2$ (убираем отрицание, получаем конъюнкцию, всё доказано), либо $p_1p_2+p_2$ (аналогично), либо $p_1p_2+p_1+p_2=$ $=(1+p_1)(1+p_2)-1=\neg(\neg p_1\wedge\neg p_2)=p_1\vee p_2$ (дизъюнкция, всё доказано). □

§3. Схемы из функциональных элементов

Формулы представляют собой способ записи композиции функций. Например, если мы сначала применяем функцию f, а потом функцию g, это можно записать формулой $g(f(x))$. Но есть и другой способ: можно изобразить каждую функцию в виде прямоугольника с "входом" и "выходом" и соединить выход функции f со входом функции g (рис. 1).

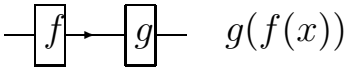


Рис. 1. Два способа изобразить композицию $g\circ f$.

Такое представление отнюдь не является чисто теоретическим. В течении нескольких десятков лет электронная промышленность выпускает микросхемы, которые выполняют логические операции. Такая микросхема имеет электрические контакты, напряжение на которых кодирует логические значения **И** и **Л**. Конкретное напряжение зависит от типа схемы, но обычно это несколько вольт, и высокий потенциал (относительно заземления) считается единицей, а низкий нулём.

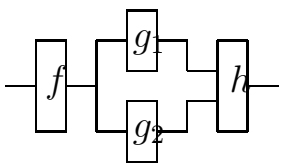
Одной из типичных схем является схема И-НЕ, она имеет два входа и один выход. Сигнал на выходе является отрицанием конъюнкции сигналов на входе. Другими словами, на выходе появляется высокий потенциал (сигнал 1) тогда и только тогда, когда на одном из входов потенциал низкий (0). Из такой схемы легко получить схему НЕ (изменяющую уровень сигнала на противоположный), соединив проводом два входа. При этом на оба входа поступает один и тот же сигнал, и операция И его не меняет ($p \wedge p = p$), а НЕ меняет на противоположный. Взяв два элемента И-НЕ и используя второй из них в качестве элемента НЕ, инвертирующего сигнал с выхода первого элемента, получаем схему, которая реализует функцию И. А если поставить два элемента НЕ перед каждым из входов элемента И-НЕ, получим схему, реализующую функцию ИЛИ: $\neg(\neg p \wedge \neg q) \leftrightarrow (p \vee q)$.

Теорема 18 о полноте системы связок теперь гарантирует, что любую булеву функцию можно реализовать в виде схемы. Надо иметь в виду, однако, что предлагаемая в её доказательстве конструкция (дизъюнктивная нормальная форма) имеет скорее теоретический интерес, поскольку приводит к схемам очень большого размера даже для простых функций (если число аргументов велико). Например, схема, сравнивающая два 16-битных числа, должна иметь 32 входа и поэтому в её реализации с помощью дизъюнктивной нормальной формы будет порядка 2^{32} элементов — что мало реально. (Между тем такую схему можно построить гораздо проще, из нескольких сотен элементов.)

Поэтому вопрос о том, сколько элементов нужно для реализации той или иной функции, представляет большой интерес — как практический, так и философский. (Одна из центральных проблем математики и информатики, так называемая "проблема перебора", может быть сформулирована в этих терминах.)

Мы сейчас дадим более формальное определение схемы и реализуемой ей булевой функции. Но прежде всего ответим на такой вопрос — почему мы вообще говорим о схемах? Ведь можно записать композицию булевых функций в виде формулы, не будет ли это то же самое?

Оказывается, не совсем, и разницу легко увидеть на примере (рис. 2).



$h(g_1(f(x)), g_2(f(x)))$

Рис. 2. Элемент входит в формулу дважды.

Здесь один и тот же элемент схемы (f) приходится указывать в формуле

дважды, поскольку его выход используется в качестве входа двух других элементов. Схемы, в которых такого ветвления нет (на практике ветвление вполне возможно, хотя и ограничено "нагрузочной способностью выхода", как говорят инженеры), как раз и соответствуют формулам. Но в общем случае полученная из данной схемы формула может быть длинной, даже если схема содержит небольшое число элементов, поскольку число копий может расти экспоненциально с ростом глубины схемы.

Хотя идея образования схемы из функциональных элементов, реализующих булевы функции, достаточно наглядна, дадим более формальное определение. Фиксируем некоторый набор булевых функций B. Пусть имеется n булевых (принимающих значения 0 и 1) переменных $x_1, \ldots, x_n$, называемых *входами*. Пусть также имеется некоторое число булевых переменных $y_1, \ldots, y_m$, называемых *проводниками*. Пусть для каждого проводника схемы задана булева функция из B, выражающая его значение через другие проводники и входы. При этом требуется, чтобы не было циклов (цикл образуется, когда y_i зависит от y_j, которое зависит от y_k, ..., которое зависит от y_i). Пусть, кроме того, среди проводников выделен один, называемый *выходом*. В таком случае говорят, что задана *схема из функциональных элементов в базисе* B с n входами. Число m называют *размером* схемы. (С точки зрения инженера размер — это число использованных элементов, а базис B — это ассортимент доступных ему элементов.)

Отсутствие циклов гарантирует, что есть проводник, зависящий только от входов (иначе можно было бы найти цикл: возьмём какой-то проводник, затем возьмём тот проводник, от которого он зависит и т. д.). Значение этого проводника, таким образом, однозначно определяется сигналами на входах. Среди оставшихся проводников также нет цикла, поэтому можно найти один из них, зависящий только от уже известных, и определить его значение. Перенумеровав проводники в таком порядке, мы можем записать последовательность присваиваний

$$\begin{aligned} y_1 &:= f_1(\ldots); \\ y_2 &:= f_2(\ldots); \\ &\ldots\ldots\ldots \\ y_m &:= f_m(\ldots), \end{aligned}$$

в правых частях которых стоят функции из B, применённые ко входам и уже найденным значениям. При этом можно считать, что результат схемы есть y_m (все последующие присваивания уже не нужны). Такая программа определяет y_m при известных значениях входов, и тем самым *вычисляет* некоторую булеву функцию.

Теперь набор булевых функций B можно назвать *полным*, если любая булева функция может быть задана схемой из B-элементов (существует программа, её вычисляющая, при этом в правых частях присваиваний стоят функции из B). Ясно, что это определение полноты равносильно прежнему, то есть возможности записать булеву функцию в виде формулы со связками из B (как мы говорили, разница только в том, что один и тот же элемент будет фигурировать в формуле многократно).

Сложностью булевой функции f относительно B называют минимальный размер схемы из B-элементов, вычисляющей функцию f. Его обозначают $\mathrm{size}_B(f)$.

Теорема 22. *Пусть B_1 и B_2 — два полных набора булевых функций. Тогда соответствующие меры сложности отличаются не более чем на постоянный множитель: найдётся такое число C, что $\mathrm{size}_{B_1}(f) \leqslant C\,\mathrm{size}_{B_2}(f)$ и $\mathrm{size}_{B_2}(f) \leqslant C\,\mathrm{size}_{B_1}(f)$ для любой функции f.*

Доказательство. Утверждение почти очевидно: поскольку наборы B_1 и B_2 полны, то каждая функция одного из наборов может быть вычислена какой-то схемой, составленной из элементов другого набора. Теперь можно взять в качестве C наибольший размер таких схем, и неравенства будут выполняться: каждую строку программы можно заменить на C (или меньше) строк с использованием функций другого набора. □

Что можно сказать о сложности произвольной булевой функции n аргументов? Следующая теорема показывает, что она экспоненциально зависит от n (для "наугад взятой" функции).

Теорема 23. *(**а**) Пусть $c > 2$. Тогда сложность любой булевой функции n аргументов не превосходит c^n для всех достаточно больших n. (**б**) Пусть $c < 2$. Тогда сложность большинства булевых функций n аргументов не меньше c^n для всех достаточно больших n.*

Доказательство. Прежде всего заметим, что по предыдущей теореме не имеет значения, какой полный базис выбрать (изменение значения c более существенно, чем умножение сложности на константу).

Первое утверждение теоремы очевидно: размер схемы, реализующей дизъюнктивную нормальную форму с n переменными, есть $O(n2^n)$, поскольку имеется не более 2^n конъюнктов размера $O(n)$. (Напомним смысл O-обозначений: $O(n2^n)$ означает, что существует верхняя оценка вида $Cn2^n$ для некоторой константы C.) Осталось заметить, что $O(n2^n) < c^n$ при достаточно больших n (напомним, что $c > 2$).

Чтобы доказать второе утверждение, оценим число различных схем (скажем, в базисе И, ИЛИ, НЕ) размера N с n аргументами. Каждая такая

схема может быть описана последовательностью из N присваиваний, выражающих одну из переменных через предыдущие. Для каждого присваивания есть не более $3(N+n)^2$ вариантов (три типа операций — конъюнкция, дизъюнкция, отрицание, и каждый из не более чем двух аргументов выбирается среди не более чем $N+n$ вариантов). Отсюда легко получить оценку $2^{O(N\log N)}$ на число всех функций сложности не более N (считая $N \geqslant n$).

Всего булевых функций с n аргументами имеется 2^{2^n}. Из сравнения этих формул видно, что что при $c < 2$ и при достаточно больших n булевы функции сложности меньше c^n составляют меньшинство, так как $2^{O(c^n \log c^n)}$ много меньше 2^{2^n}. □

Задача 101. *Проведите вторую часть рассуждения более подробно и покажите, что при некотором $\varepsilon > 0$ сложность большинства булевых функций с n аргументами не меньше $\varepsilon 2^n/n$.*

Верхнюю оценку в теореме 23 можно усилить и показать, что сложность любой булевой функции n аргументов не превосходит $O(2^n/n)$.

Задача 102. *(**а**) Покажите, что возможно построить схему размера $O(2^m)$ с 2^m выходами, реализующую все 2^m возможных конъюнктов длины m (для каждого — свой выход). (Указание: такую схему можно построить индуктивно.) (**б**) Покажите, что можно построить схему размера $O(2^{2^m})$ с 2^{2^m} выходами, реализующую все 2^{2^m} булевых функций m аргументов. (Указание: эту схему также можно построить индуктивно.) (**в**) Пусть $\varphi(x_1,\ldots,x_k,y_1,\ldots,y_l)$ — булева функция, аргументы которой разбиты на две группы. Покажите, что её можно записать в виде дизъюнкции 2^k членов, каждый из которых имеет вид $C(x_1,\ldots,x_k) \wedge \wedge D(y_1,\ldots,y_l)$, где C — конъюнкт, а D — произвольная булева функция. Вывести отсюда упомянутую выше оценку $O(2^n/n)$. (Указание: разумно положить $k = n - \log n + c$, $l = \log n - c$.)*

Теорема 23, однако, ничего не говорит о сложности конкретных булевых функций. Ситуация здесь такова. Есть разнообразные методы и приёмы получения верхних оценок. Но про нижние оценки неизвестно практически ничего. Про многие функции мы подозреваем, что их сложность велика (экспоненциально зависит от числа входов), но доказать это пока не удаётся. Весьма нетривиальные идеи позволяют доказывать экспоненциальные нижние оценки для некоторых специальных классов схем, например, схем из монотонных элементов или схем ограниченной глубины (использующих элементы И и ИЛИ с произвольным числом входов). Получение экспоненциальных оценок для более общих схем — один из возможных подходов к

знаменитой *проблеме перебора*, центральной проблеме теории сложности вычислений.

Мы не будем углубляться в эту теорию, а приведём лишь несколько верхних оценок для конкретных задач. При этом мы не претендуем на полноту, а хотим лишь показать несколько интересных идей и приёмов.

Рассмотрим функцию сравнения двух n-битовых чисел. Она имеет $2n$ аргументов (n для одного числа и n для другого); её значение равно 1, если первое число больше второго, и 0 в противном случае.

Обозначим эту функцию Comp_n.

Теорема 24. *Пусть B — полный набор функций. Существует такая константа C, что* $\mathrm{size}_B(\mathrm{Comp}_n) \leqslant Cn$.

Доказательство. Заметим, что поскольку в формулировке теоремы оценка размера проводится с точностью до константы, то выбор конкретного базиса не имеет значения. Другими словами, мы можем предполагать, что любое конечное число необходимых нам функций в этом базисе есть.

Схема сравнения чисел будет рекурсивной (чтобы сравнить два числа, мы отдельно сравниваем их левые и правые половины, а затем объединяем результаты). При этом, как часто бывает, надо усилить утверждение, чтобы индукция прошла. А именно, мы будем строить схему с $2n$ входами $x_1, \ldots, x_n, y_1, \ldots, y_n$ и двумя выходами, которая указывает, какой из трёх случаев $x < y$, $x = y$ или $x > y$ имеет место. (Здесь x, y — числа, записываемое в двоичной системе как $x_1 \ldots x_n$ и $y_1, \ldots, y_n$.) Два выходных бита кодируют четыре возможности, а нужно только три, так что есть некоторый запас. Для определённости можно считать, что первый выходной бит истинен, если числа равны, а второй — если $x < y$. Тогда возможны три варианта сигналов на выходе: 10 (равенство), 01 (при $x < y$) и 00 (при $x > y$).

Объясним теперь, как собрать, скажем, схему сравнения двух 16-разрядных чисел. Соберём отдельно схему сравнения старших 8 разрядов и младших 8 разрядов. Каждая из них даст ответ в форме двух битов. Теперь из этих четырёх битов надо собрать два. (Если в старших разрядах неравенство, то оно определяет результат сравнения; если старшие разряды равны, то результат сравнения определяется младшими разрядами.) Написанная в скобках фраза определяет булеву функцию с четырьмя битами на входе и двумя битами на выходе, и может быть реализована некоторой схемой фиксированного размера. Таким образом, если через $T(n)$ обозначить размер схемы, сравнивающей n-битовые числа, то получаем оценку $T(2n) \leqslant 2T(n) + c$, где c — некоторая константа, зависящая от выбора базиса. Отсюда следует, что $T(2^k) \leqslant c'2^k$ при некотором c'. В самом деле, для

достаточно большого c' можно доказать по индукции, что $T(2^k) \leqslant c'2^k - c$ (мы должны усилить неравенство, вычтя из правой части c, чтобы индуктивный шаг прошёл; база индукции остается верной, если c' достаточно велико).

Ту же самую оценку можно объяснить и наглядно. Наша схема имеет вид иерархического дерева. На каждом уровне из двух двухбитовых сигналов получается один. Остаётся вспомнить, что в полном двоичном дереве число внутренних вершин (которое определяет размер схемы) на единицу меньше числа листьев. (В турнире по олимпийской системе число игр на единицу меньше числа команд, так как после каждой игры одна команда выбывает.)

Все внутренние вершины и все листья (где сравниваются два бита) представляют собой схемы ограниченного размера, откуда и вытекает оценка $T(2^k) \leqslant c'2^k$.

Осталось лишь сказать, что делать, если размер чисел (который мы обозначали через n) не есть точная степень двойки. В этом случае можно увеличить размер до ближайшей сверху степени двойки (не более чем в два раза) и подать на старшие разряды входов нули. Оба действия приводят к увеличению размера схемы не более чем в константу раз. □

Перейдём к сложению двух n-разрядных чисел. (Строго говоря, тут возникает не булева функция, а функция $\mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}^{n+1}$, но все наши определения очевидно переносятся на этот случай.)

Теорема 25. *Существует схема размера $O(n)$, осуществляющая сложение двух n-битовых чисел.*

Доказательство. Напомним смысл обозначения $O(n)$: нам надо построить схему сложения n-битовых чисел, имеющую размер не более cn для некоторого c и для всех n.

Вспомним, как складывают числа в столбик:

$$\begin{array}{r} 011 \\ 1001 \\ 1011 \\ \hline 10100 \end{array}$$

Верхняя строка — биты переноса, нижняя — результат. Заметим, что каждый из битов переноса или результата определяется тремя другими битами (бит результата равен сумме двух битов слагаемых и бита переноса по модулю 2, а бит переноса равен 1, если хотя бы два из этих трёх битов равны 1). Поэтому можно составить схему, которая вычисляет эти биты справа налево и имеет размер $O(n)$. □

Заметим, что теорему 24 легко вывести из теоремы 25: чтобы сравнить числа x и y, сложим число $(2^n - 1) - x$ (то есть число x, в котором все единицы заменены нулями и наоборот) и число y. Если в старшем разряде появится единица, то $y > x$, а если нет, то $y \leqslant x$. Остаётся заметить, что и сложение, и обращение битов в числе x требуют схем линейного размера. Таким образом, сравнение чисел сводится к вычислению бита переноса. Верно и обратное: вычисление бита переноса сводится к сравнению двух чисел (обратим в одном из слагаемых все биты).

Тем не менее конструкция, использованная при доказательстве теоремы 24, имеет некоторые преимущества. Назовём *глубиной* схемы максимальное число элементов на пути от входа к выходу. Если представить себе, что сигнал на выходе элемента появляется не сразу после подачи сигналов на входы, а с некоторой задержкой, то глубина схемы определяет суммарную задержку. Легко понять, что рекурсивная схема сравнения имела глубину $O(\log n)$ (число уровней пропорционально логарифму размера входа), в то время как построенная только что схема сложения имеет глубину, пропорциональную n (биты переноса вычисляются последовательно, справа налево). Но можно соединить эти два результата:

Теорема 26. *Существует схема сложения двух n-битовых чисел размера $O(n)$ и глубины $O(\log n)$.*

Доказательство. Как мы видели, проблема в том, что биты переноса вычисляются последовательно, а не параллельно. Если удастся их все вычислить схемой размера $O(n)$ и глубины $O(\log n)$, то дальнейшее очевидно.

Как мы упоминали, вычисление битов переноса равносильно сравнению, так что для доказательства теоремы достаточно научиться сравнивать параллельно все "суффиксы" двух n-битовых чисел $x_1 \dots x_n$ и $y_1 \dots y_n$, т. е. для каждого i сравнить числа $x_i x_{i+1} \dots x_n$ и $y_i y_{i+1} \dots y_n$.

Вспомним, что мы делали при сравнении чисел (скажем, длины 8). На нижнем уровне мы сравнивали биты:

$$\begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \end{array}$$

На следующем уровне мы сравнивали двузначные числа

$$\begin{array}{cccc} x_1x_2 & x_3x_4 & x_5x_6 & x_7x_8 \\ y_1y_2 & y_3y_4 & y_5y_6 & y_7y_8 \end{array}$$

затем четырёхзначные

$$\begin{array}{cc} x_1x_2x_3x_4 & x_5x_6x_7x_8 \\ y_1y_2y_3y_4 & y_5y_6y_7y_8 \end{array}$$

и, наконец, восьмизначные:

$$x_1x_2x_3x_4x_5x_6x_7x_8$$
$$y_1y_2y_3y_4y_5y_6y_7y_8$$

Таким образом, для суффиксов длины 8, 4, 2 и 1 результаты сравнения уже есть. Для суффикса длины 6 результат можно получить, комбинируя результат сравнения $x_3x_4 \quad y_3y_4$ и $x_5x_6x_7x_8 \quad y_5y_6y_7y_8$. После этого у нас есть информация о суффиксах всех чётных длин, и соединяя её с информацией с первого этапа, получаем сведения про все суффиксы. Например, для сравнения суффиксов длины 7, то есть $x_2 \dots x_8$ и $y_2 \dots y_8$, мы соединяем результаты сравнения x_2 и y_2 с результатами сравнения суффиксов длины 6, то есть $x_3 \dots x_8$ и $y_3 \dots y_8$.

В общем случае картина такая: после "сужающегося дерева" мы строим "расширяющееся"; за k шагов до конца мы знаем результаты сравнения всех суффиксов, длины которых кратны 2^k. Это дерево имеет размер $O(n)$ и глубину $O(\log n)$, что завершает доказательство. □

Задача 103. *Покажите, что вычитание двух n-битовых чисел по модулю 2^n выполняется схемой размера $O(n)$ и глубины $O(\log n)$. (Указание: вычитание легко сводится к сложению, если заменить нули на единицы и наоборот.)*

Теперь займёмся умножением. Схема умножения двух n-разрядных чисел имеет $2n$ входов (по n для каждого множителя) и $2n$ выходов для произведения.

Посмотрим, какие оценки даёт обычный способ умножения чисел столбиком. В нём умножение двух n-разрядных чисел сводится к сложению n копий первого числа (частично заменённых на нули в зависимости от цифр второго числа) со сдвигами.

Получение этих копий требует схемы размера $O(n^2)$ (общее число цифр в копиях) и глубины $O(1)$. Сложение двух $2n$-разрядных чисел мы можем выполнить с помощью схемы размера $O(n)$ и глубины $O(\log n)$, так что необходимые $n-1$ сложений можно выполнить схемой размера $O(n^2)$ и глубины $O(\log^2 n)$ (если складывать сначала попарно, потом результаты снова попарно и т. д.). Оказывается, этот результат можно улучшить. Наиболее экономные способы основаны на преобразовании Фурье . С их помощью, например, можно построить схему умножения n-битовых чисел, имеющую размер $n \log^c n$.

Эти методы далеко выходят за рамки нашего обсуждения, но два улучшения мы приведём.

Теорема 27. *Существует схема умножения двух n-разрядных чисел размера $O(n^2)$ и глубины $O(\log n)$.*

Доказательство. Как мы уже говорили, умножение двух n-разрядных чисел сводится к сложению n таких чисел, и остаётся выполнить такое сложение схемой размера $O(n^2)$ и глубины $O(\log n)$. Ключевым моментом здесь является сведение сложения трёх чисел к сложению двух с помощью простой схемы размера $O(n)$ и глубины $O(1)$. В самом деле, пусть есть три числа x, y и z. Если мы будем складывать отдельно в каждом разряде, то в разряде может накопиться любая сумма от 0 до 3, то есть в двоичной записи от 00 до 11. Сформируем из младших битов этих двухбитовых сумм число u, а из старших (сдвинутых влево) — число v. Тогда, очевидно, $x + y + z = u + v$. Получение цифр числа u и v происходит параллельно во всех разрядах и требует размера $O(n)$ и глубины $O(1)$.

Теперь, если надо сложить n чисел, можно разбить их на тройки и из каждых трёх чисел получить по два. В следующий круг, таким образом, выйдут $(2/3)n$ чисел (примерно — граничные эффекты большой роли не играют). Их снова можно сгруппировать по тройкам и т. д. С каждым уровнем число слагаемых убывает в полтора раза, так что глубина схемы будет логарифмической. Каждое преобразование трёх слагаемых в два требует схемы размера $O(n)$ и уменьшает число слагаемых на единицу, так что потребуется n таких преобразований. Итак, эта конструкция имеет общий размер $O(n^2)$ и глубину $O(\log n)$. Надо только отметить, что в конце у нас получается не одно число, а два, и их напоследок надо сложить — что мы умеем делать с глубиной $O(\log n)$ и размером $O(n)$. □

Задача 104. *Докажите, что схема, вычисляющая булеву функцию f от n аргументов, у которой ни один аргумент не является фиктивным, имеет размер не менее cn и глубину не менее $c \log n$, где $c > 0$ — некоторая константа, зависящая от выбранного набора элементов. (Аргумент функции называют* фиктивным*, если от него значение функции не зависит.)*

Эта задача показывает, что если в процессе умножения двух n-разрядных чисел мы суммируем n слагаемых размера n, то оценки $O(n^2)$ для размера и $O(\log n)$ для глубины, полученные при доказательстве теоремы 27, существенно улучшить нельзя.

Однако никто не обязывает нас следовать традиционному способу умножения столбиком — отказавшись от него, мы можем уменьшить размер схемы.

Теорема 28. *Существует схема умножения двух n-разрядных чисел размера $O(n^{\log_2 3})$ и глубины $O(\log^2 n)$.*

Доказательство. Начнём с такого замечания. Вычисляя произведение двух комплексных чисел

$$(a+bi)(c+di) = (ac-bd)+(ad+bc)i$$

обычным способом, мы делаем четыре умножения. Но можно обойтись и тремя с помощью такого трюка: вычислить ac, bd и $(a+b)(c+d)$, а потом найти $ad+bc$ как разность $(a+b)(c+d)-ac-bd$.

Аналогичный фокус можно проделать и для целых чисел. Разобьём $2n$-битовое число на две n-битовые части, то есть представим его в виде $a2^n+b$. Теперь запишем произведение двух таких чисел:

$$(a2^n+b)(c2^n+d) = ac2^{2n}+(ad+bc)2^n+bd.$$

Теперь видно, что достаточно найти три произведения ac, bd и $(a+b)(c+d)$, чтобы определить все три слагаемых в правой части равенства. Получается, что умножение двух $2n$-разрядных чисел сводится к трём умножениям n-разрядных и к нескольким сложениям и вычитаниям. (На самом деле при умножении $(a+b)$ на $(c+d)$ сомножители могут быть $(n+1)$-разрядными, но это не страшно, так как обработка лишнего разряда сводится к нескольким сложениям.)

Для размера схемы, таким образом, получается рекурсивная оценка

$$S(2n) \leqslant 3S(n)+O(n),$$

из которой следует, что $S(n) = O(n^{\log_2 3})$. В самом деле, для умножения n-разрядных чисел требуется дерево рекурсивных вызовов глубины $\log_2 n$ и степени ветвления 3. Заметим, что размер схемы в вершине пропорционален числу складываемых битов. При переходе от одного уровня к следующему (более близкому к корню) размер слагаемых растёт вдвое, а число вершин уменьшается втрое, поэтому общее число элементов на этом уровне уменьшается в полтора раза. Таким образом, при движении по уровням от листьев к корню получается убывающая геометрическая прогрессия со знаменателем $2/3$, сумма которой всего лишь втрое превосходит её первый член. Остаётся заметить, что число листьев равно $3^{\log_2 n} = n^{\log_2 3}$.

Оценка глубины также очевидна: на каждом уровне мы имеем схему сложения глубины $O(\log n)$, а число уровней есть $O(\log n)$. □

На этом мы завершаем знакомство со схемами из функциональных элементов, выполняющими арифметические операции. Рассмотрим теперь

функцию "голосования"Она имеет нечётное число аргументов, и значение её равно 0 или 1 в зависимости от того, какое из двух значений чаще встречается среди входов.

Теорема 29. *Существует схема, вычисляющая функцию голосования, размера $O(n)$ и глубины $O(\log n \log\log n)$.*

Доказательство. На самом деле можно даже вычислить общее число единиц среди входов. Это делается рекурсивно: считаем отдельно для каждой половины, потом складываем. Получается логарифмическое число уровней. На верхнем уровне надо складывать числа размера $\log n$, на следующем — размера $(\log n - 1)$ и так до самого низа, где складываются однобитовые числа (то есть биты входа). Какой средний размер складываемых чисел? Половина вершин в дереве приходится на нижний уровень (числа длины 1), четверть — на следующий (числа длины 2) и т. д. Вспоминая, что ряд $\sum(k/2^k)$ сходится, видим, что средний размер складываемых чисел есть $O(1)$ и общий размер схемы есть $O(n)$. А общая глубина есть $O(\log n \log\log n)$, так как на каждом из $\log n$ уровней стоит схема глубины $O(\log\log n)$. □

Заметим, что хотя функция голосования монотонна, построенная схема её вычисления содержит немонотонные элементы (поскольку операция сложения не монотонна). Мы уже говорили, что всякую монотонную функцию можно составить из конъюнкций и дизъюнкций. Для функции голосования есть очевидный способ это сделать: написать дизъюнкцию всех конъюнкций размера $(n+1)/2$ (напомним, что число входов n предполагается нечётным). Однако при этом получится схема экспоненциального по n размера.

Теорема 30. *Существует схема размера $O(n^c)$ и глубины $O(\log n)$, составленная только из элементов И и ИЛИ (с двумя входами), вычисляющая функцию голосования.*

Доказательство. Для начала заметим, что ограничение на размер является следствием ограничения на глубину, так как элементы И и ИЛИ имеют только два входа и число элементов в схеме глубины d есть $O(2^d)$.

Схема будет строиться из элементов большинства с тремя входами. (Каждый из них можно собрать из конъюнкций и дизъюнкций по формуле $(a\wedge b)\vee(a\wedge c)\vee(b\wedge c)$.) Выход схемы будет большинством из трёх значений, каждое из которых есть большинство из трёх значений и т. д. (рис. 3).

Продолжая эту конструкцию на k уровнях, мы получим схему с 3^k входами. (Отметим, что эта схема не будет вычислять большинство среди своих входов — по той же причине, по которой результат непрямого голосования может отличаться от мнения большинства.) Но мы сделаем вот какую

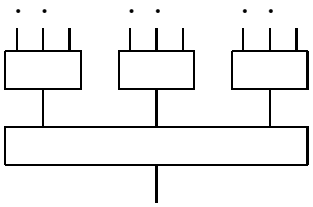


Рис. 3. Дерево из элементов 3-большинства.

странную вещь: возьмём k равным $c \log n$ при достаточно большом коэффициенте пропорциональности c (при этом число входов такой схемы будет полиномиально зависеть от n) и напишем на входах случайно выбранные переменные из данного нам набора $x_1, \ldots, x_n$. (Переменные, записываемые на разных входах, выбираются независимо.) Оказывается, что с ненулевой вероятностью эта схема будет вычислять функцию большинства среди $x_1, \ldots, x_n$, если константа c достаточно велика. Следовательно, искомая схема существует.

Обратите внимание: нам удастся доказать существование интересующей нас схемы, не предъявив её явно. (Такое использование вероятностных методов в комбинаторных рассуждениях часто бывает полезно.)

Итак, почему же схема с положительной вероятностью вычисляет функцию большинства? Это доказывается так: рассмотрим какой-то один набор значений на входах и докажем, что на этом конкретном наборе случайная схема выдаёт правильный ответ с вероятностью, очень близкой к единице (равной $1 - \varepsilon$ при очень малом ε).

Если число ε настолько мало, что остаётся меньшим единицы даже после умножения на число возможных входов (2^n), то получаем требуемое (каждое из 2^n событий имеет вероятность не меньше $1 - \varepsilon$, значит их пересечение имеет вероятность не меньше $1 - 2^n\varepsilon > 0$).

Итак, осталось оценить вероятность того, что случайная схема даст правильный ответ на данном входе. Пусть доля единиц среди всех входов равна p. Тогда на каждый входной провод схемы подаётся единица с вероятностью p и нуль с вероятностью $1 - p$ (выбор случайной переменной даёт единицу с вероятностью p), причём сигналы на всех входах независимы.

Если на трёх входах элемента 3-большинства сигналы независимы, и вероятность появления единицы на каждом входе есть p, то вероятность появления единицы на выходе есть $\varphi(p) = 3p^2(1-p) + p^3 = 3p^2 - 2p^3$. На следующих уровнях вероятность появления единицы будет равна $\varphi(\varphi(p)), \varphi(\varphi(\varphi(p))), \ldots$ График функции $\varphi(x)$ на отрезке $[0, 1]$ (рис. 3) показывает, что при итерациях функции φ дисбаланс (отклонение от середины) нарастает и последовательность стремится к краю отрезка. Надо только оценить число шагов.

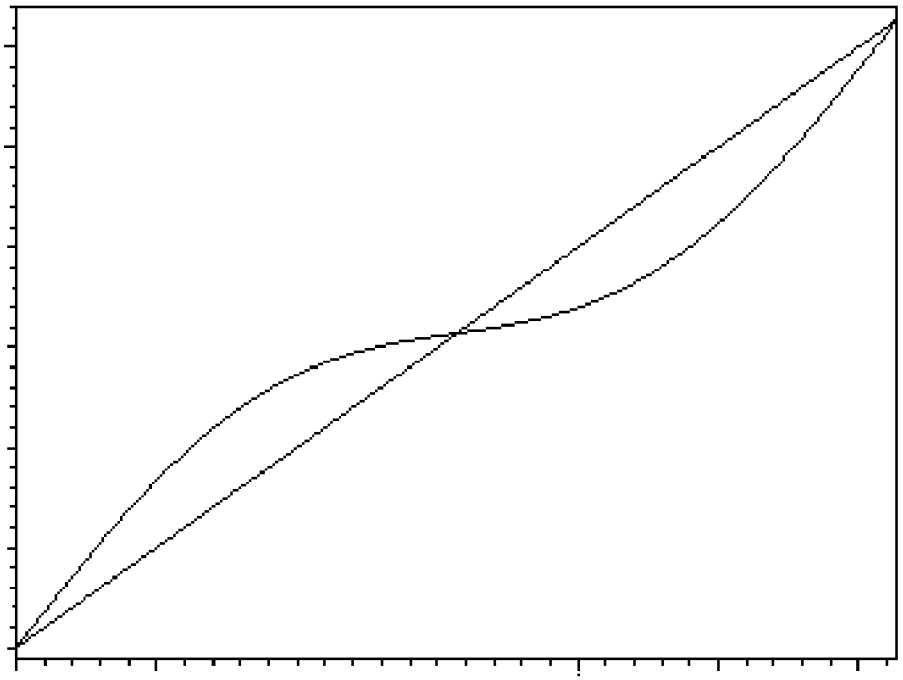


Рис. 4. Итерируемая функция φ.

Если вначале единицы составляют большинство из n аргументов (напомним, n нечётно), то их как минимум $(n+1)/2$, так что $p \geqslant (n+1)/2n = 1/2 + 1/(2n)$. Таким образом, начальный дисбаланс составляет как минимум $1/2n$. А в конце нам нужно приблизиться к краю отрезка на расстояние 2^{-n}.

Итак, нам осталось доказать такую лемму (относящуюся скорее к математическому анализу):

Лемма. Пусть последовательность $x_k \in [0,1]$ задана рекуррентной формулой $x_{k+1} = \varphi(x_k)$, где

$$\varphi(x) = 3x^2 - 2x^3.$$

Пусть $x_0 \geqslant 1/2 + 1/(2n)$. Тогда последовательность x_k монотонно возрастает и приближается к 1 на расстояние 2^{-n} за $O(\log n)$ шагов. [Симметричное утверждение верно и при $x_0 \leqslant 1/2 - 1/(2n)$.]

Идея доказательства: посмотрим на функцию вблизи точки $1/2$ и у краёв отрезка. В точке $1/2$ производная больше 1, поэтому удаление от $1/2$ растёт как геометрическая прогрессия, и точка перейдёт какую-то фиксированную границу (например, 0,51) не позднее чем за $O(\log n)$ шагов. Затем потребуется $O(1)$ шагов, чтобы дойти, скажем, до 0,99. В единице первая производная функции равна нулю, поэтому расстояние до единицы каждый раз примерно возводится в квадрат, и потому для достижения погрешности 2^{-n} потребуется $O(\log n)$ шагов (как в методе Ньютона отыскания корня). Всего получается $O(\log n) + O(1) + O(\log n)$ шагов, что и требовалось. $\square$

На самом деле справедливо гораздо более сильное утверждение: существует схема размера $O(n \log n)$ и глубины $O(\log n)$, состоящая только из элементов И и ИЛИ, которая имеет n входов и n выходов и осуществляет сортировку последовательности n нулей и единиц (это означает, что на выходе столько же единиц, сколько на входе, причём выходная последовательность всегда невозрастающая). Ясно, что средний бит выхода в такой ситуации реализует функцию большинства.

При кажущейся простоте формулировки единственная известная конструкция такой схемы (сортирующая сеть AKS, придуманная Айтаи, Комлошом и Сцемереди сравнительно недавно, в 1983 году) весьма сложна, и появление какой-то более простой конструкции было бы замечательным достижением.

ГЛАВА IV

Исчисление высказываний

Напомним, что тавтологией мы называли пропозициональную формулу, истинную при всех значениях переменных. Оказывается, что все тавтологии можно получить из некоторого набора "аксиом" с помощью "правил вывода", которые имеют чисто синтаксический характер и никак не апеллируют к смыслу формулы, её истинности и т. д. Эту задачу решает так называемое *исчисление высказываний (ИВ)*. В этой главе мы перечислим аксиомы и правила вывода этого исчисления, и приведём несколько доказательств *теоремы о полноте* (которая утверждает, что всякая тавтология выводима в исчислении высказываний).

§1. Исчисление высказываний

Каковы бы ни были формулы A, B, C, следующие формулы называют *аксиомами исчисления высказываний*:

(1) $A \to (B \to A)$;
(2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;
(3) $(A \wedge B) \to A$;
(4) $(A \wedge B) \to B$;
(5) $A \to (B \to (A \wedge B))$;
(6) $A \to (A \vee B)$;
(7) $B \to (A \vee B)$;
(8) $(A \to C) \to ((B \to C) \to (A \vee B \to C))$;
(9) $\neg A \to (A \to B)$;
(10) $(A \to B) \to ((A \to \neg B) \to \neg A)$;
(11) $A \vee \neg A$.

Как говорят, мы имеем здесь одиннадцать "схем аксиом"; из каждой схемы можно получить различные конкретные аксиомы, заменяя входящие в неё буквы на пропозициональные формулы.

Единственным правилом вывода исчисления высказываний является правило со средневековым названием "modus ponens" (MP). Это правило разрешает получить (вывести) из формул A и $(A \to B)$ формулу B.

Выводом в исчислении высказываний называется конечная последовательность формул, каждая из которых есть аксиома или получается из предыдущих по правилу modus ponens.

Вот пример вывода (в нём первая формула является частным случаем схемы (1), вторая — схемы (2), а последняя получается из двух предыдущих по правилу modus ponens):

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

Пропозициональная формула A называется *выводимой* в исчислении высказываний, или *теоремой* исчисления высказываний, если существует вывод, в котором последняя формула равна A. Такой вывод называют выводом формулы A. (В принципе можно было бы и не требовать, чтобы формула A была последней — все дальнейшие формулы можно просто вычеркнуть.)

Как мы уже говорили, в исчислении высказываний выводятся все тавтологии и только они. Обычно это утверждение разбивают на две части: простую и сложную. Начнём с простой:

Теорема 31 (о корректности ИВ). *Всякая теорема исчисления высказываний есть тавтология.*

Доказательство. Несложно проверить, что все аксиомы — тавтологии. Для примера проделаем это для самой длинной аксиомы (точнее, схемы аксиом) — для второй. В каком случае формула

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

(где A, B, C — некоторые формулы) могла бы быть ложной? Для этого посылка $A \to (B \to C)$ должна быть истинной, а заключение $(A \to B) \to (A \to C)$ — ложным. Чтобы заключение было ложным, формула $A \to B$ должна быть истинной, а формула $A \to C$ — ложной. Последнее означает, что A истинна, а C ложна. Таким образом, мы знаем, что A, $(A \to B)$ и $(A \to (B \to C))$ истинны. Отсюда следует, что B и $(B \to C)$ истинны, и потому C истинна — противоречие. Значит, наша формула не бывает ложной.

Корректность правила MP также очевидна: если формулы $(A \to B)$ и A всегда истинны, то по определению импликации формула B также всегда истинна. Таким образом, все формулы, входящие в выводы (все теоремы) являются тавтологиями. □

Гораздо сложнее доказать обратное утверждение.

Теорема 32 (о полноте ИВ). *Всякая тавтология есть теорема исчисления высказываний.*

Доказательство. Мы предложим ряд альтернативных доказательств этой теоремы. Но прежде всего мы должны приобрести некоторый опыт построения выводов и использования аксиом.

Лемма 1. Какова бы ни была формула D, формула $(D \to D)$ является теоремой.

Докажем лемму, предъявив вывод формулы $(D \to D)$ в исчислении высказываний.

1. $(D \to ((D \to D) \to D)) \to ((D \to (D \to D)) \to (D \to D))$
   [аксиома 2 при $A = D$, $B = (D \to D)$, $C = D$];
2. $D \to ((D \to D) \to D)$ [аксиома 1];
3. $(D \to (D \to D)) \to (D \to D)$ [из 1 и 2 по правилу MP];
4. $D \to (D \to D)$ [аксиома 1];
5. $(D \to D)$ [из 3 и 4 по правилу MP].

Как видно, вывод даже такой простой тавтологии, как $(D \to D)$, требует некоторой изобретательности. Мы облегчим себе жизнь, доказав некоторое общее утверждение о выводимости.

Часто мы рассуждаем так: предполагаем, что выполнено какое-то утверждение A, и выводим различные следствия. После того как другое утверждение B доказано, мы вспоминаем, что использовали предположение A, и заключаем, что мы доказали утверждение $A \to B$. Следующая лемма, называемая иногда "леммой о дедукции", показывает, что этот подход правомерен и для исчисления высказываний.

Пусть Γ — некоторое множество формул. *Выводом из* Γ называется конечная последовательность формул, каждая из которых является аксиомой, принадлежит Γ или получается из предыдущих по правилу MP. (Другими словами, мы как бы добавляем формулы из Γ к аксиомам исчисления высказываний — именно как формулы, а не как схемы аксиом.) Формула A *выводима из* Γ, если существует вывод из Γ, в котором она является последней формулой. В этом случае мы пишем $\Gamma \vdash A$. Если Γ пусто, то речь идёт о выводимости в исчислении высказываний, и вместо $\varnothing \vdash A$ пишут просто $\vdash A$.

Лемма 2 (о дедукции). Пусть Γ — множество формул. Тогда $\Gamma \vdash A \to B$ тогда и только тогда, когда $\Gamma \cup \{A\} \vdash B$.

В одну сторону утверждение почти очевидно: пусть $\Gamma \vdash (A \to B)$. Тогда и $\Gamma, A \vdash (A \to B)$. (Для краткости мы опускаем фигурные скобки и заменяем знак объединения запятой.) По определению $\Gamma, A \vdash A$, откуда по MP получаем $\Gamma, A \vdash B$.

Пусть теперь $\Gamma, A \vdash B$. Нам надо построить вывод формулы $A \to B$ из Γ. Возьмём вывод $C_1, C_2, \ldots, C_n$ формулы $B = C_n$ из Γ, A. Припишем ко всем

формулам этого вывода слева посылку A:

$$(A \to C_1), (A \to C_2), \ldots, (A \to C_n).$$

Эта последовательность оканчивается на $(A \to B)$. Сама по себе она не будет выводом из Γ, но из неё можно получить такой вывод, добавив недостающие формулы, и тем самым доказать лемму о дедукции.

Будем добавлять эти формулы, двигаясь слева направо. Пусть мы подошли к формуле $(A \to C_i)$. По предположению формула C_i либо совпадает с A, либо принадлежит Γ, либо является аксиомой, либо получается из двух предыдущих по правилу MP. Рассмотрим все эти случаи по очереди.

(1) Если C_i есть A, то очередная формула имеет вид $(A \to A)$. По лемме 1 она выводима, так что перед ней мы добавляем её вывод.

(2) Пусть C_i принадлежит Γ. Тогда мы вставляем формулы C_i и $C_i \to (A \to C_i)$ (аксиома 1). Применение правила MP к этим формулам даёт $(A \to C_i)$, что и требовалось.

(3) Те же формулы можно добавить, если C_i является аксиомой исчисления высказываний.

(4) Пусть, наконец, формула C_i получается из двух предыдущих формул по правилу MP. Это значит, что в исходном выводе ей предшествовали формулы C_j и $(C_j \to C_i)$. Тогда в новой последовательности (с добавленной посылкой A) уже были формулы $(A \to C_j)$ и $(A \to (C_j \to C_i))$. Поэтому мы можем продолжить наш Γ-вывод, написав формулы

$((A \to (C_j \to C_i)) \to ((A \to C_j) \to (A \to C_i))$ (аксиома 2);
$((A \to C_j) \to (A \to C_i))$ (modus ponens);
$(A \to C_i)$ (modus ponens).

Итак, во всех четырёх случаях мы научились дополнять последовательность до вывода из Γ, так что лемма о дедукции доказана.

Задача 105. *Докажите, что для любых формул A, B, C формула*

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

выводима в исчислении высказываний. (Указание: используйте лемму о дедукции и тот факт, что $A \to B, B \to C, A \vdash C$.)

Задача 106. *Докажите, что если $\Gamma_1 \vdash A$ и $\Gamma_2, A \vdash B$, то $\Gamma_1 \cup \Gamma_2 \vdash B$. (Это свойство иногда называют "правилом сечения" (cut); говорят, что формула A "отсекается" или "высекается". Сходные правила играют центральную роль в теории доказательств, где формулируется и доказывается "теорема об устранении сечения" для различных логических систем.)*

Задача 107. *Добавим к исчислению высказываний, помимо правила modus ponens, ещё одно правило, называемое* правилом подстановки. *Оно разрешает заменить в выведенной формуле все переменные на произвольные формулы (естественно, вхождения одной переменной должны заменяться на одну и ту же формулу). Покажите, что после добавления такого правила класс выводимых формул не изменится, но теорема о дедукции перестанет быть верной.*

Заметим, что мы пока что использовали только две первые аксиомы исчисления высказываний. Видно, кстати, что они специально подобраны так, чтобы доказательство леммы о дедукции прошло.

Другие аксиомы описывают свойства логических связок. Аксиомы 3 и 4 говорят, какие следствия можно вывести из конъюнкции ($A \wedge B \vdash A$ и $A \wedge B \vdash B$). Напротив, аксиома 5 говорит, как можно вывести конъюнкцию. Из неё легко следует такое правило: если $\Gamma \vdash A$ и $\Gamma \vdash B$, то $\Gamma \vdash (A \wedge B)$ (применяем эту аксиому и дважды правило MP). Часто подобные правила записывают так:

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

(над чертой пишут "посылки" правила, а снизу — его "заключение", вытекающее из посылок).

Задача 108. *Докажите, что формула* $(A \to (B \to C)) \to ((A \wedge B) \to \to C)$, *так же как и обратная к ней формула (в которой посылка и заключение переставлены), являются теоремами исчисления высказываний. Докажите аналогичное утверждение про формулы* $(A \wedge B) \to (B \wedge A)$ *и* $((A \wedge B) \wedge C) \to (A \wedge (B \wedge C))$.

Аксиомы 6 – 7 позволяют утверждать, что $A \vdash A \vee B$ и $B \vdash A \vee B$. Аксиома 8 обеспечивает такое правило:

$$\frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}$$

Оно соответствует такой схеме рассуждения: "Пусть выполнено $A \vee B$. Разберём два случая. Если выполнено A, то $\langle \ldots \rangle$ и потому C. Если выполнено B, то $\langle \ldots \rangle$ и потому C. В обоих случаях верно C. Значит, $A \vee B$ влечёт C." Обоснование: дважды воспользуемся леммой о дедукции, получив $\Gamma \vdash (A \to \to C)$ и $\Gamma \vdash (B \to C)$, а затем дважды применим правило MP к этим формулам и аксиоме $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$. Получив формулу $(A \vee B) \to C$, опять применим правило MP к ней и формуле $(A \vee B)$.

Задача 109. *Докажите, что следующие формулы, а также обратные к ним (меняем местами посылку и заключение) являются теоремами исчисления высказываний:*

$$((A \vee B) \to C) \to ((A \to C) \wedge (B \to C)),$$
$$((A \wedge C) \vee (B \wedge C)) \to ((A \vee B) \wedge C),$$
$$((A \vee C) \wedge (B \vee C)) \to ((A \wedge B) \vee C).$$

У нас остались ещё три аксиомы, касающиеся отрицания. Аксиома 9 гарантирует, что из противоречивого набора посылок можно вывести что угодно: если $\Gamma \vdash A$ и $\Gamma \vdash \neg A$, то $\Gamma \vdash B$ для любого B. Аксиома 10, напротив, объясняет, как можно вывести отрицание некоторой формулы A: надо допустить A и вывести два противоположных заключения B и $\neg B$. Точнее говоря, имеет место такое правило:

$$\frac{\Gamma, A \vdash B \qquad \Gamma, A \vdash \neg B}{\Gamma \vdash \neg A}$$

(в самом деле, дважды применяем лемму о дедукции, а затем правило MP с аксиомой 10).

Аксиомы 9 и 10 позволяют вывести некоторые логические законы, связанные с отрицанием. Докажем, например, что (для любых формул A и B) формула

$$(A \to B) \to (\neg B \to \neg A)$$

("закон контрапозиции") является теоремой исчисления высказываний. В самом деле, по лемме о дедукции достаточно установить, что

$$(A \to B), \neg B \vdash \neg A.$$

Для этого, в свою очередь, достаточно вывести из посылок $(A \to B), \neg B, A$ какую-либо формулу и её отрицание (в данном случае формулы B и $\neg B$).

Задача 110. *Выведите формулы $A \to \neg\neg A$ и $\neg\neg\neg A \to \neg A$ с помощью аналогичных рассуждений.*

Последняя аксиома, называемая "законом исключённого третьего", и иногда читаемая как "третьего не дано" (tertium non datur в латинском оригинале), вызвала в первой половине века большое количество споров. (В интуиционистской логике этой аксиомы нет.)

Из неё можно вывести закон "снятия двойного отрицания",имеющий вид $\neg\neg A \to A$. В самом деле, достаточно показать, что $A \vee \neg A, \neg\neg A \vdash A$. По правилу разбора случаев, достаточно установить, что $A, \neg\neg A \vdash A$ (это очевидно) и что $\neg A, \neg\neg A \vdash A$ (а это верно, так как из двух противоречащих друг другу формул выводится что угодно с помощью аксиомы 8).

Задача 111. *Докажите, что формула* $(\neg B \to \neg A) \to (A \to B)$ *является теоремой исчисления высказываний. (Указание: используйте закон исключённого третьего.)*

Задача 112. *Исключим из числа аксиом исчисления высказываний закон исключённого третьего, заменив его на закон снятия двойного отрицания. Покажите, что от этого класс выводимых формул не изменится.*

Задача 113. *Докажите, что при наличии аксиомы исключённого третьего (11) аксиома (10) является лишней — её (точнее следовало бы сказать: любой частный случай этой схемы аксиом) можно вывести из остальных аксиом.*

Теперь уже можно доказать теорему о полноте: всякая тавтология выводима в исчислении высказываний. Идея доказательства состоит в разборе случаев. Поясним её на примере. Пусть A — произвольная формула, содержащая переменные p, q, r. Предположим, что A истинна, когда все три переменные истинны. Тогда, как мы докажем,

$$p, q, r \vdash A.$$

Вообще каждой строке таблицы истинности для формулы A соответствует утверждение о выводимости. Например, если A ложна, когда p и q ложны, а r истинно, то

$$\neg p, \neg q, r \vdash \neg A.$$

Если формула A является тавтологией, то окажется, что она выводима из всех восьми возможных вариантов посылок. Пользуясь законом исключённого третьего, можно постепенно избавляться от посылок. Например, из $p, q, r \vdash A$ и $p, q, \neg r \vdash A$ можно получить $p, q, (r \vee \neg r) \vdash A$, то есть $p, q \vdash A$ (поскольку $(r \vee \neg r)$ является аксиомой).

Проведём это рассуждение подробно. Для начала докажем такую лемму:

Лемма 3. Для произвольных формул P и Q

$$\begin{aligned}
P, Q &\vdash (P \wedge Q); & P, Q &\vdash (P \vee Q);\\
P, \neg Q &\vdash \neg(P \wedge Q); & P, \neg Q &\vdash (P \vee Q);\\
\neg P, Q &\vdash \neg(P \wedge Q); & \neg P, Q &\vdash (P \vee Q);\\
\neg P, \neg Q; &\vdash \neg(P \wedge Q) & \neg P, \neg Q &\vdash \neg(P \vee Q);
\end{aligned}$$

$$\begin{aligned}
P, Q &\vdash (P \to Q);\\
P, \neg Q &\vdash \neg(P \to Q); & P &\vdash \neg(\neg P);\\
\neg P, Q &\vdash (P \to Q); & \neg P &\vdash \neg P.\\
\neg P, \neg Q &\vdash (P \to Q);
\end{aligned}$$

Эта лемма говорит, что если принять в качестве гипотез истинность или ложность формул P и Q, являющихся частями конъюнкции, дизъюнкции или импликации, то можно будет доказать или опровергнуть всю формулу (в зависимости от того, истинна она или ложна). Последняя часть содержит аналогичное утверждение про отрицание.

После предпринятой нами тренировки доказать эти утверждения несложно. Например, убедимся, что $\neg P \vdash \neg(P \wedge Q)$. Для этого достаточно вывести два противоположных утверждения из $\neg P, (P \wedge Q)$ — ими будут утверждения P и $\neg P$.

Проверим ещё одно утверждение: $\neg P, \neg Q \vdash \neg(P \vee Q)$. Нам надо вывести два противоположных утверждения из $\neg P, \neg Q, (P \vee Q)$. Покажем, что из $\neg P, \neg Q, (P \vee Q)$ следует всё, что угодно. По правилу разбора случаев достаточно убедиться, что из $\neg P, \neg Q, P$ и из $\neg P, \neg Q, Q$ следует всё, что угодно — но это мы знаем.

Утверждения, касающиеся импликации, просты: в самом деле, мы знаем, что $Q \vdash (P \to Q)$ благодаря аксиоме 1, а $\neg P \vdash (P \to Q)$ благодаря аксиоме 9.

Остальные утверждения леммы столь же просты.

Теперь мы можем сформулировать утверждение о разборе случаев для произвольной формулы.

Лемма 4. Пусть A — произвольная формула, составленная из переменных $p_1, \ldots, p_n$. Тогда для каждой строки таблицы истинности формулы A имеет место соответствующее утверждение о выводимости: если $\varepsilon_1, \ldots, \varepsilon_n, \varepsilon \in \in \{0, 1\}$, и значение формулы A есть ε при $p_1 = \varepsilon_1, \ldots, p_n = \varepsilon_n$, то

$$\neg_{\varepsilon_1} p_1, \ldots, \neg_{\varepsilon_n} p_n \vdash \neg_{\varepsilon} A,$$

где $\neg_u \varphi$ обозначает φ при $u = 1$ и $\neg\varphi$ при $u = 0$ (напомним, что 1 обозначает истину, а 0 — ложь).

Лемма очевидно доказывается индукцией по построению формулы A. Мы имеем посылки, утверждающие истинность или ложность переменных, и для всех подформул (начиная с переменных и идя ко всей формуле) выводим их или их отрицания с помощью леммы 3.

Если формула A является тавтологией, то из всех 2^n вариантов посылок выводится именно она, а не её отрицание. Тогда правило разбора случаев и закон исключённого третьего позволяют избавиться от посылок: сгруппируем их в пары, отличающиеся в позиции p_1 (в одном наборе посылок стоит p_1, в другом $\neg p_1$), по правилу разбора случаев заменим их на посылку $(p_1 \vee \neg p_1)$, которую можно выбросить (она является аксиомой). Сделав так для всех пар, получим 2^{n-1} выводов, в посылках которых нет p_1; повторим этот процесс с посылками p_2, $\neg p_2$ и т. д. В конце концов мы убедимся, что формула A выводима без посылок, как и утверждает теорема о полноте. $\square$

§2. Второе доказательство теоремы о полноте

Это доказательство, в отличие от предыдущего, обобщается на более сложные случаи (исчисление предикатов, интуиционистское исчисление высказываний). Начнём с такого определения: множество формул Γ называется *совместным*, если существует набор значений переменных, при которых все формулы из Γ истинны. Заметим, что формула φ является тавтологией тогда и только тогда, когда множество, состоящее из единственной формулы $\neg\varphi$, не является совместным. Для случая одной формулы есть специальный термин: формула τ *выполнима*, если существуют значения переменных, при которых она истинна, то есть если множество $\{\tau\}$ совместно. Тавтологии — это формулы, отрицания которых не выполнимы.

Множество формул Γ называется *противоречивым*, если из него одновременно выводятся формулы A и $\neg A$. Мы знаем, что в этом случае из него выводятся вообще все формулы, см 1. (В противном случае Γ называется *непротиворечивым*.)

Теорема 33 (корректность исчисления высказываний, вторая форма). *Всякое совместное множество формул непротиворечиво.*

Доказательство. В самом деле, пусть совместное множество Γ противоречиво. Так как оно совместно, существуют значения переменных, при которых все формулы из Γ истинны. С другой стороны, из Γ выводится некоторая формула B и её отрицание. Может ли так быть?

Оказывается, что нет. Мы уже видели, что всякая выводимая формула истинна при всех значениях переменных (является тавтологией). Справедливо и несколько более общее утверждение: если $\Gamma \vdash A$ и при некоторых значениях переменных все формулы из Γ истинны, то и формула A истинна при этих значениях переменных. (Как и раньше, это легко доказывается индукцией по построению вывода A из Γ.)

В нашей ситуации это приводит к тому, что на выполняющем наборе значений переменных для Γ должны быть истинны обе формулы B и $\neg B$, что, разумеется, невозможно. □

Мы называем это утверждение другой формой теоремы о корректности исчисления высказываний, поскольку из него формально можно вывести, что всякая теорема является тавтологией: если A — теорема, то множество $\{\neg A\}$ противоречиво (из него выводятся A и $\neg A$), потому несовместно, значит, $\neg A$ всегда ложна, поэтому A всегда истинна.

Теорема 34 (полнота исчисления высказываний, вторая форма). *Любое непротиворечивое множество совместно.*

Доказательство. Нам дано непротиворечивое множество Γ, а надо найти такие значения переменных, при которых все формулы из Γ истинны. (Вообще говоря, множество Γ может быть бесконечно и содержать бесконечное число разных переменных.)

Пусть есть какая-то переменная p, встречающаяся в формулах из семейства Γ. Нам надо решить, сделать ли её истинной или ложной. Если оказалось так, что из Γ выводится формула p, то выбора нет: она обязана быть истинной в тех наборах, где формулы из Γ истинны (как мы видели при доказательстве корректности). По тем же причинам, если из Γ выводится $\neg p$, то в выполняющем наборе переменная p обязательно будет ложной.

Если оказалось так, что для любой переменной p либо она сама, либо её отрицание выводятся из Γ, то выполняющий набор значений определён однозначно, и надо только проверить, что он действительно будет выполняющим. А если для каких-то переменных нельзя вывести ни их, ни их отрицание, то мы пополним наш набор Γ так, чтобы они, как теперь модно говорить, "определились".

Проведём это рассуждение подробно. Рассмотрим все переменные, входящие в какие-либо формулы из множества Γ; обозначим множество этих переменных через V. Зафиксируем это множество и до конца доказательства теоремы о полноте будем рассматривать только формулы с переменными из множества V, не оговаривая этого особо.

Назовём непротиворечивое множество Γ *полным*, если для любой формулы F имеет место либо $\Gamma \vdash F$, либо $\Gamma \vdash \neg F$ (одновременно этого быть не может, так как Γ непротиворечиво).

Утверждение теоремы о полноте очевидно следует из двух лемм:

Лемма 1. Всякое непротиворечивое множество Γ содержится в непротиворечивом полном множестве Δ.

Лемма 2. Для всякого непротиворечивого полного множества Δ существует набор значений переменных (из V, напомним), при котором все формулы из Δ истинны.

Доказательство леммы 1. Основную роль здесь играет такое утверждение: если Γ — непротиворечивое множество, а A — произвольная формула, то хотя бы одно из множеств $\Gamma \cup \{A\}$ и $\Gamma \cup \{\neg A\}$ непротиворечиво. В самом деле, если оба множества $\Gamma \cup \{A\}$ и $\Gamma \cup \{\neg A\}$ противоречивы, то $\Gamma \vdash \neg A$ и $\Gamma \vdash \neg\neg A$, но множество Γ предполагалось непротиворечивым.

Если множество переменных V конечно или счётно, то доказательство леммы 1 легко завершить: множество всех формул тогда счётно, и просматривая их по очереди, мы можем добавлять к Γ либо саму формулу, либо её отрицание, сохраняя непротиворечивость. Получится, очевидно, полное множество. Чуть менее очевидна его непротиворечивость: оно было непро-

тиворечиво на каждом шаге, но почему предельное множество (объединение возрастающей последовательности) будет непротиворечиво? Дело в том, что в выводе двух противоречащих друг другу формул может быть задействовано только конечное число формул из Γ (по определению выводимости: вывод есть конечная последовательность формул). Поэтому все эти формулы должны появиться на некотором конечном шаге конструкции, а это невозможно (на всех шагах множество непротиворечиво).

Для случая произвольного набора переменных V рассуждение можно завершить ссылкой на лемму Цорна: рассмотрим частично упорядоченное множество, элементами которого будут непротиворечивые множества формул, а порядком — отношение "быть подмножеством". Рассуждение предыдущего абзаца показывает, что всякая цепь в этом множестве имеет верхнюю границу (объединение линейно упорядоченного по включению семейства непротиворечивых множеств является непротиворечивым множеством). Следовательно, для любого непротиворечивого множества найдётся содержащее его максимальное непротиворечивое множество. А оно обязано быть полным (иначе его можно расширить, добавив A или $\neg A$).

Лемма 1 доказана.

Доказательство леммы 2. Пусть Γ — непротиворечивое полное множество. Тогда для каждой переменной (из множества V) ровно одна из формул p и $\neg p$ выводима из Γ. Если первая, будем считать переменную p истинной, если вторая — ложной. Тем самым появляется некоторый набор ν значений переменных, и надо только проверить, что любая формула из Γ при таких значениях переменных истинна. Это делается так: индукцией по построению формулы A мы доказываем, что

$$A \text{ истинна на наборе } \nu \;\Rightarrow\; \Gamma \vdash A,$$
$$A \text{ ложна на наборе } \nu \;\Rightarrow\; \Gamma \vdash \neg A.$$

Базис индукции (когда A — переменная) обеспечивается определением истинности переменных. Для шага индукции используется та же лемма, что и при доказательстве полноты с помощью разбора случаев. Пусть, например, A имеет вид $(B \wedge C)$. Тогда есть четыре возможности для истинности B и C. В одном из них (когда B и C истинны на ν) по предположению индукции мы имеем $\Gamma \vdash B$ и $\Gamma \vdash C$, откуда $\Gamma \vdash (B \wedge C)$, то есть $\Gamma \vdash A$. В другом (B истинна, C ложна) предположение индукции даёт $\Gamma \vdash B$ и $\Gamma \vdash \neg C$, откуда $\Gamma \vdash \neg(B \wedge C)$, то есть $\Gamma \vdash \neg A$. Аналогично разбираются и все остальные случаи и логические связки. Лемма 2 доказана, и тем самым завершено доказательство теоремы 34. □

Мы доказали, что всякое непротиворечивое множество формул совмест-

но. Отсюда легко следует, что всякая тавтология является теоремой. В самом деле, если φ — тавтология, то множество $\{\neg\varphi\}$ несовместно, поэтому из $\neg\varphi$ выводится противоречие, поэтому $\vdash \neg\neg\varphi$, и по закону снятия двойного отрицания $\vdash \varphi$.

Кроме того, теорема о полноте во второй формулировке имеет такое очевидное следствие:

Теорема 35 (теорема компактности для исчисления высказываний). *Пусть* Γ *— множество формул, всякое конечное подмножество которого совместно. Тогда и всё множество* Γ *совместно.*

Доказательство. Как мы знаем, несовместность равносильна противоречивости, а вывод противоречия по определению может использовать лишь конечное число формул. □

§3. О женской логике

Как известно[1], великая русская литература предсказала многое - от атомной бомбы ("Мир рвался в опытах Кюри атóмной лопнувшею бомбой" - это у Андрея Белого) до формализации женской логики (это у Лермонтова).

Женская психология интересовала едва ли не всех русских писателей, женская логика — лишь избранных. Если брать только классиков, то прямые заявления на этот счёт можно найти у Тургенева и у Лермонтова. Тургенев устами Пигасова (в "Рудине", гл. 2) заявляет: "... мужчина может, например, сказать, что дважды два не четыре, а пять или три с половиною, а женщина скажет, что дважды два — стеариновая свечка". Придирчивый критик заметит, что здесь скорее говорится не о какой-то там женской логике, а о том, что женщина склонна к высказываниям, лежащим вне всякой логики. Лермонтов демонстрирует более тонкий подход. Устами (а точнее рукою) Печорина он стремится проанализировать характерные для женской логики структуры рассуждения. Вот запись из журнала Печорина от 11-го июня:

> Нет ничего парадоксальнее женского ума:
> порядок доказательств, которым они уничтожают свои предубеждения, очень оригинален; чтобы выучиться их диалектике, надо опрокинуть в уме своём все школьные правила логики.
>
> Например, способ обыкновенный: Этот человек любит меня, но я замужем, следовательно, не должна его любить.

[1]Ниже пересказывается раздел из книги "Труды по не математике"крупного специалиста по математической логике В. А. Успенского.

Способ женский: Я не должна его любить, ибо я замужем; но он меня любит, — следовательно...

Тут несколько точек, ибо рассудок уже ничего не говорит...

Придирчивый критик и тут не найдёт того опрокидывания всех правил логики, на которое ссылается Печорин. Скорее, скажет этот критик, здесь вступают в конфликт два силлогизма, нравственный и чувственный, и чувственный побеждает. (Сформулируем оба для ясности. Нравственный силлогизм: замужняя женщина не должна любить никого, кроме своего мужа; он — не мой муж, а я замужем; следовательно, я не должна его любить. Чувственный силлогизм: я люблю того, кто любит меня; он меня любит; следовательно, я его люблю.) Критику мы возразим, что слово *следовательно* во фразе, избранной Печориным для иллюстрирования женского способа, не вполне уместно после двух предшествующих ему посылок: *Я не должна его любить, ибо я замужем* и *Он меня любит*; из этих посылок по правилам обычной логики мало что следует. Кроме того, возразим мы критику, преобладание гедонистического начала, которое прослеживается в печоринском примере, а ещё точнее — использование этого начала в качестве важнейшего элемента логической конструкции и есть одна из характерных черт женской логики. Это было установлено Колмогоровым.

"При имени Колмогорова тотчас осеняет мысль о русском национальном учёном", как сказал бы Гоголь. Поэтому разговор о русской литературе плавно перетекает в разговор о русской науке. Среди представителей российской науки есть те, кого по общемировым стандартам можно без колебаний назвать великим учёным. Таковы, на наш взгляд, трое: Михаил Васильевич Ломоносов, Дмитрий Иванович Менделеев, Андрей Николаевич Колмогоров. Уже для весьма нами уважаемого Ивана Петровича Павлова мы предпочли бы понятие 'великий физиолог'.

Для полноты картины надо привести здесь некоторые факты научной биографии Колмогорова Андрея Николаевича (25.04.1903 – 20.10.1987). Логика была любовью его молодости; он вернулся к ней на склоне своих лет. В 1925 г. Колмогоров опубликовал статью "О принципе tertium non datur", входящую в общепризнанный золотой фонд сочинений по математической логике, сочинений, определивших лицо этой науки. А с начала 1980 г. до конца жизни Колмогоров возглавлял кафедру математической логики Московского университета.

Колмогоровская статья 1925 г. была посвящена так называемой интуиционистской логике, а именно — её формализации. *Интуиционистская* логика, в отличие от обычной, называемой также *классической*, не признаёт закона исключённого третьего, он же — принцип "третьего не дано"

(tertium non datur). Этот принцип утверждает, что какое ни возьми высказывание **A**, что-нибудь одно, **A** или **не- A**, непременно верно: не может быть, чтобы было верно нечто третье. Формализация же какой-либо логики, будь то классической, интуиционистской или иной, состоит в том, что предъявляется два точно описанных и исчерпывающих списка: список *аксиом* и список *правил вывода*. Аксиомы провозглашаются истинными по определению; например, в классической (но, разумеется, не в интуиционистской) логике в качестве одной из аксиом как раз и выступает закон исключённого третьего. Правила вывода задают те процедуры, посредством которых из заданных посылок выводятся непосредственные следствия; верны или неверны при этом сами посылки, несущественно. Одним из правил вывода (и для классической, изложенной в этой главе, и для интуиционистской логики) является, например, такое:

Из двух посылок: [**если P, то Q**] *и* **P** — *следует* **Q**.

Или, короче,

Пусть [**P** ⇒ **Q**] *и* **P**; *тогда* **Q**.

Это правило называют правилом modus ponens.

Всё сказанное имело целью подготовить читателя к восприятию колмогоровского открытия. Колмогоров обнародовал своё правило в 80-х годах. Открытие состоит в формулировке следующего правила женской логики:

Правило Колмогорова: *Пусть* [**P** ⇒ **Q**] *и* [**Q приятно**]; *тогда* **P**.

Колмогоров не утрудил себя приведением какого-либо примера. Приведём таковой для ясности.

Итак, вот пример на применение правила Колмогорова: *если у мужа есть деньги, у меня будет новая шубка* (это есть **P** ⇒ **Q**); *иметь новую шубку приятно* (это есть **Q приятно**); отсюда (по правилу Колмогорова) следует, что *у мужа есть деньги* (это есть **P**).

ГЛАВА V

Языки первого порядка

Помимо логических связок, в математических рассуждениях часто встречаются *кванторы* "для любого" ($\forall$) и "существует" ($\exists$). Например, определение непрерывности начинается словами "для любого положительного ε найдётся положительное δ, для которого ...". А одна из аксиом теории групп (существование обратного элемента) записывается так: $\forall x \exists y((xy = = 1) \wedge (yx = 1))$.

Можно сформулировать различные логические законы, включающие в себя кванторы. Например, высказывание "существует такое x, что A" (где A — некоторое свойство объекта x) логически эквивалентно высказыванию "не для всех x верно $\neg A$".

Мы будем записывать такого рода законы с помощью формул, дадим определение истинности формул (при данной интерпретации входящих в них символов) и исследуем, какого рода свойства можно выражать с помощью формул и какие нельзя.

§1. Формулы и интерпретации

Начнём с примера. Пусть M — некоторое непустое множество, а R — бинарное отношение на нём, то есть подмножество декартова произведения $M \times M$. Вместо $\langle x, y\rangle \in R$ мы будем писать $R(x, y)$. Рассмотрим формулу

$$\forall x\, \exists y\, R(x, y).$$

Эта формула выражает некоторое свойство бинарного отношения R (для любого элемента $x \in M$ найдётся элемент, находящийся с ним в отношении R) и может быть истинна или ложна. Например, если M есть множество натуральных чисел $\mathbb{N}$, а R — отношение "строго меньше" (другими словами, R есть множество всех пар $\langle x, y\rangle$, для которых $x < y$), то эта формула истинна. А для отношения "строго больше" (на том же множестве) эта формула ложна.

Вопрос о том, будет ли истинна формула

$$\exists y\, R(x, y)$$

для данного множества M и для данного бинарного отношения R на нём, не имеет смысла, пока не уточнено, каково значение переменной x. Например, если $M = \mathbb{N}$ и $R(x, y)$ есть $x > y$, то эта формула будет истинной при $x = 3$

и ложной при $x = 0$. Для данных M и R она задаёт некоторое свойство элемента x и тем самым определяет некоторое подмножество множества M.

Перейдём к формальным определениям. Пусть M — непустое множество. Множество M^k состоит из всех последовательностей $\langle m_1, \ldots, m_k \rangle$ длины k, составленных из элементов множества M. Назовём *k-местной функцией* на множестве M любое отображение M^k в M (определённое на всём M^k). Синонимы: "функция k аргументов", "функция валентности k", "функция местности k" и даже "функция арности k" (последнее слово происходит от слов "унарная" для функций одного аргумента, "бинарная" (операция) для функций двух аргументов и "тернарная" для трёх аргументов).

Назовём *k-местным предикатом* на множестве M любое отображение M^k в множество $\mathbb{B} = \{\textbf{И}, \textbf{Л}\}$. Такой предикат будет истинным на некоторых наборах $\langle m_1, \ldots, m_k \rangle$ множества M и ложным на остальных наборах. Поставив ему в соответствие множество тех наборов, где он истинен, мы получаем взаимно однозначное соответствие между k-местными предикатами на M и подмножествами множества M^k. Говоря о предикатах, также употребляют термины "валентность", "число аргументов" и др.

Мы будем рассматривать также функции и предикаты валентности нуль. Множество M^0 одноэлементно (содержит единственную последовательность длины 0). Поэтому функции $M^0 \to M$ отождествляются с элементами множества M, а нульместных предикатов ровно два — истинный и ложный.

Естественно, что в формулы будут входить не сами функции и предикаты, а обозначения для них, которые называют *функциональными* и *предикатными символами*. В качестве символов можно использовать любые знаки. Важно лишь, что каждому символу приписана валентность, которая определяет, со сколькими аргументами он может встречаться в формуле. Произвольный набор предикатных и функциональных символов, для каждого из которых указано неотрицательное число, называемое *валентностью*, мы будем называть *сигнатурой*.

Остаётся определить три вещи: что такое формула данной сигнатуры, что такое интерпретация данной сигнатуры и когда формула является истинной (в данной интерпретации).

Фиксируем некоторый набор символов, называемых *индивидными переменными*. Они предназначены для обозначения элементов множества, на котором определены функции и предикаты; обычно в таком качестве используют латинские буквы x, y, z, u, v, w с индексами. В каждой формуле будет использоваться конечное число переменных, так что счётного набора переменных нам хватит. Мы предполагаем, что переменные отличны от всех функциональных и предикатных символов сигнатуры (иначе выйдет

путаница).

Определим понятие *терма* данной сигнатуры. Термом называется последовательность переменных, запятых, скобок и символов сигнатуры, которую можно построить по следующим правилам:

- Индивидная переменная есть терм.
- Функциональный символ валентности 0 есть терм.
- Если $t_1, \ldots, t_k$ — термы, а f — функциональный символ валентности $k > 0$, то $f(t_1, \ldots, t_k)$ есть терм.

В принципе можно было не выделять функциональные символы валентности 0 (которые также называют *константами*) в отдельную группу, но тогда бы после них пришлось писать скобки (как это делается в программах на языке Си).

Если A — предикатный символ валентности k, а $t_1, \ldots, t_k$ — термы, то выражение $A(t_1, \ldots, t_k)$ считается *атомарной формулой*. Кроме того, любой предикатный символ валентности 0 считается атомарной формулой.

Формулы строятся по таким правилам:

- Атомарная формула есть формула.
- Если φ — формула, то $\neg\varphi$ — формула.
- Если φ и ψ — формулы, то выражения $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ также являются формулами.
- Если φ есть формула, а ξ — индивидная переменная, то выражения $\forall\xi\,\varphi$ и $\exists\xi\,\varphi$ являются формулами.

Во многих случаях в сигнатуру входит двуместный предикатный символ $=$, называемый *равенством*. По традиции вместо $=(t_1, t_2)$ пишут $(t_1 = t_2)$.

Итак, понятие формулы в данной сигнатуре полностью определено. Иногда такие формулы называют *формулами первого порядка* данной сигнатуры, или формулами *языка первого порядка* с данной сигнатурой.

Наш следующий шаг — определение *интерпретации* данной сигнатуры. Пусть фиксирована некоторая сигнатура σ. Чтобы задать интерпретацию сигнатуры σ, необходимо:

- указать некоторое непустое множество M, называемое *носителем* интерпретации;
- для каждого предикатного символа сигнатуры σ указать предикат с соответствующим числом аргументов, определённый на множестве M (как мы уже говорили, 0-местным предикатным символам ставится в соответствие либо **И**, либо **Л**);
- для каждого функционального символа сигнатуры σ указать функцию соответствующего числа аргументов с аргументами и значениями из M (в частности, для 0-местных функциональных символов

надо указать элемент множества M, с ними сопоставляемый).

Если сигнатура включает в себя символ равенства, то среди её интерпретаций выделяют *нормальные* интерпретации, в которых символ равенства интерпретируется как совпадение элементов множества M.

Приведём несколько примеров сигнатур, используемых в различных теориях.

Сигнатура теории упорядоченных множеств включает в себя два двуместных предикатных символа (равенство и порядок) и не имеет функциональных символов. Здесь также вместо $\leqslant(x, y)$ по традиции пишут $x \leqslant y$.

Аксиомы порядка (рефлексивность, антисимметричность, транзитивность) могут быть записаны формулами этой сигнатуры. Например, требование антисимметричности записывается так:

$$\forall x\, \forall y(((x \leqslant y) \wedge (y \leqslant x)) \to (x = y)).$$

Иногда в сигнатуру теории упорядоченных множеств вместо символа $\leqslant$ включают символ $<$; большой разницы тут нет.

Задача 114. *Как записать с помощью формулы свойство линейной упорядоченности? свойство не иметь наибольшего элемента? свойство плотности (отсутствия соседних элементов)? свойство фундированности (отсутствия бесконечных убывающих последовательностей — или, что эквивалентно, наличия минимального элемента в любом подмножестве)? свойство полной упорядоченности? (Указание: не для всех перечисленных свойств это возможно.)*

Сигнатуру теории групп можно выбирать по-разному. Можно считать, что (помимо равенства) она имеет двуместный функциональный символ $\times$ (который по традиции записывают между множителями), константу (нульместный функциональный символ) 1 и одноместный функциональный символ $\mathrm{inv}(x)$ для обращения. Тогда аксиомы теории групп записываются с использованием лишь кванторов всеобщности:

$$\forall x\, \forall y\, \forall z(((x \times y) \times z) = (x \times (y \times z))),$$
$$\forall x\, (((x \times 1) = x) \wedge ((1 \times x) = x)),$$
$$\forall x\, (((x \times \mathrm{inv}(x)) = 1) \wedge ((\mathrm{inv}(x) \times x) = 1)).$$

Если не включать операцию обращения в сигнатуру, придётся использовать квантор существования и переписать последнюю аксиому так:

$$\forall x\, \exists y\, (((x \times y) = 1) \wedge ((y \times x) = 1)).$$

Задача 115. *Как записать аксиомы теории групп, если в сигнатуре нет константы* 1*? (Указание: аксиома о существовании обратного станет частью аксиомы о существовании единицы.)*

Задача 116. *Как записать в виде формулы требование коммутативности группы? утверждение о том, что любой элемент (кроме единицы) имеет порядок* 11*? конечность группы? (Указание: не всё из перечисленного можно записать, хотя пока у нас нет средств это установить.)*

Сигнатура теории множеств содержит два двуместных предикатных символа: для принадлежности и для равенства. Аксиомы теории множеств можно записывать в виде формул этой сигнатуры. Чаще всего рассматривают вариант аксиоматической теории множеств, называемый теорией Цермело – Френкеля и обозначаемый ZF. Приведём для примера одну из аксиом теории ZF, называемую *аксиомой объёмности*, или *экстенсиональности*:

$$\forall x\,\forall y\,((\forall z\,((z\in x)\to(z\in y))\wedge\forall z\,((z\in y)\to(z\in x)))\to(x=y)).$$

Задача 117. *Сформулировать словесно эту аксиому.*

Задача 118. *Какова естественная сигнатура для теории полей? Можно ли записать в виде формулы этой сигнатуры утверждение о том, что поле имеет характеристику* 2*? конечную характеристику? алгебраически замкнуто?*

Задача 119. *Докажите основные эквивалентности, содержащие кванторы.*

1) $\overline{\forall x\ P(x)}\equiv\exists x\ \overline{P(x)}$
2) $\overline{\exists x\ P(x)}\equiv\forall x\ \overline{P(x)}$
3) $\forall x\forall y\ P(x,y)\equiv\forall y\forall x\ P(x,y)$
4) $\exists x\exists y\ P(x,y)\equiv\exists y\exists x\ P(x,y)$
5) $\forall x\ (P(x)\wedge Q(x))\equiv\forall x\ P(x)\wedge\forall x\ Q(x))$
6) $\exists x\ (P(x)\vee Q(x))\equiv\exists x\ P(x)\vee\exists x\ Q(x))$
7) $\forall x\ (P(x)\vee Q(y))\equiv\forall x\ P(x)\vee Q(y)$
8) $\exists x\ (P(x)\wedge Q(y))\equiv\exists x\ P(x)\wedge Q(y)$
9) $\exists y\forall x\ P(x,y)\Rightarrow\forall x\exists y\ P(x,y)\equiv 1$

§2. Определение истинности

Из приведённых выше примеров, вероятно, понятен смысл формулы, то есть ясно, в каких интерпретациях данной сигнатуры и для каких элементов формула истинна. Тем не менее для любителей строгости мы приведём

формальное определение истинности. (Его детали понадобятся, когда мы будем проверять истинность выводимых формул, см. раздел 3.)

Прежде всего, определим формально понятие *параметра* формулы (переменной, от значения которой может зависеть истинность формулы). Согласно этому определению, скажем, формула $\forall x\, \exists y\, A(x,y)$ не имеет параметров, а формулы $\exists y\, A(x,y)$ и $(A(x) \wedge \forall x\, B(x,x))$ имеют единственный параметр x. Вот как выглядит это определение:

- Параметрами терма являются все входящие в него индивидные переменные.
- Параметрами атомарной формулы являются параметры всех входящих в неё термов.
- Параметры формулы $\neg\varphi$ те же, что у формулы φ.
- Параметрами формул $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ и $(\varphi \to \psi)$ являются все параметры формулы φ, а также все параметры формулы ψ.
- Параметрами формул $\forall\xi\, \varphi$ и $\exists\xi\, \varphi$ являются все параметры формулы φ, кроме переменной ξ.

Параметры иногда называют *свободными переменными* формулы. Заметим, что формула может иметь одновременно параметр x и использовать (в другом месте) квантор $\forall x$. Как говорят в этом случае, одна и та же переменная имеет *свободные*и *связанные* вхождения. Свободное вхождение переменной — это такое вхождение, которое не входит в область действия одноимённого квантора. Если аккуратно определить эту область действия, несложно проверить, что параметры формулы — это как раз переменные, имеющие свободные вхождения.

Теперь мы хотим определить понятие формулы, истинной в данной интерпретации при данных значениях параметров. Технически проще считать, что всем индивидным переменным приписаны какие-то значения, а потом доказать, что переменные, не являющиеся параметрами, не влияют на истинность формулы.

Итак, пусть фиксирована сигнатура и некоторая интерпретация этой сигнатуры. *Оценкой* назовём отображение, которое ставит в соответствие каждой индивидной переменной некоторый элемент множества, являющегося носителем интерпретации. Этот элемент будем называть *значением переменной* при данной оценке.

Определим индуктивно *значение терма* t при данной оценке π, которое мы будем обозначать $[t](\pi)$.

- Для переменных оно уже определено.
- Если t является константой (нульместным функциональным символом), то $[t](\pi)$ не зависит от π и равно значению этой константы при

данной интерпретации (напомним, в интерпретации с каждой константой сопоставляется некоторый элемент носителя).

- Если t имеет вид $f(t_1, \ldots, t_m)$, где f — функциональный символ валентности m, а $t_1, \ldots, t_m$ — термы, то $[t](\pi)$ есть $[f]([t_1](\pi), \ldots, [t_m](\pi))$, где $[f]$ есть функция, соответствующая символу f в нашей интерпретации, а $[t_i](\pi)$ есть значение терма t_i при оценке π.

Теперь можно определить *значение формулы* φ при данной оценке π в данной интерпретации, которое обозначается $[\varphi](\pi)$ и может быть равно **И** или **Л**; в первом случае формула называется *истинной*, во втором — *ложной*. Это определение также индуктивно:

- Значение атомарной формулы $A(t_1, \ldots, t_m)$ определяется как

$$[A]([t_1](\pi), \ldots, [t_m](\pi)),$$

где $[A]$ — предикат, соответствующий предикатному символу A в рассматриваемой интерпретации. Если формула представляет собой нульместный предикатный символ, то её значение не зависит от оценки и есть значение этого символа.
- $[\neg\varphi](\pi)$ определяется как $\neg[\varphi(\pi)]$, где $\neg$ понимается как операция в $\mathbb{B}$. Другими словами, формула $\neg\varphi$ истинна при оценке π тогда и только тогда, когда формула φ ложна при этой оценке.
- $[\varphi \wedge \psi](\pi)$ определяется как $[\varphi](\pi) \wedge [\psi](\pi)$, где $\wedge$ в правой части понимается как операция в $\mathbb{B}$. (Другими словами, формула $(\varphi \wedge \psi)$ истинна при оценке π тогда и только тогда, когда обе формулы φ и ψ истинны при этой оценке.) Аналогичным образом $[\varphi \vee \psi](\pi)$ определяется как $[\varphi](\pi) \vee [\psi](\pi)$, а $[\varphi \to \psi](\pi)$ — как $[\varphi](\pi) \to [\psi](\pi)$.
- Формула $\forall\xi\, \varphi$ истинна на оценке π тогда и только тогда, когда формула φ истинна на любой оценке π', которая совпадает с π всюду, кроме значения переменной ξ (которое в оценке π' может быть любым). Другими словами, если обозначить через $\pi + (\xi \mapsto m)$ оценку, при которой значение переменной ξ равно m, а остальные переменные принимают те же значения, что и в оценке π, то

$$[\forall\xi\, \varphi](\pi) = \bigwedge_{m \in M} [\varphi](\pi + (\xi \mapsto m)).$$

(В правой части стоит бесконечная конъюнкция, которая истинна, если все её члены истинны.)
- Формула $\exists\xi\, \varphi$ истинна на оценке π тогда и только тогда, когда формула φ истинна на некоторой оценке π', которая совпадает с π всюду,

кроме значения переменной ξ (которое в оценке π' может быть любым). Другими словами,

$$[\forall \xi\, \varphi](\pi) = \bigvee_{m \in M} [\varphi](\pi + (\xi \mapsto m)).$$

(В правой части стоит бесконечная дизъюнкция, которая истинна, если хотя бы один из её членов истинен.)

Заметим, что в двух последних пунктах значение переменной ξ в оценке π не играет роли. Это позволяет легко доказать (индукцией по построению формулы) такое утверждение: если две оценки π_1 и π_2 придают одинаковые значения всем параметрам формулы φ, то $[\varphi](\pi_1) = [\varphi](\pi_2)$. Другими словами, истинность формулы определяется значениями её параметров.

Задача 120. *Проведите это индуктивное рассуждение подробно.*

Задача 121. *Приведённые выше определения применимы к любой формуле, в том числе и к странной формуле $\forall y\, A(x)$. Какие у неё параметры? При каких значениях параметров она истинна? (Ответ: она имеет единственный параметр x и эквивалентна формуле $A(x)$.)*

Задача 122. *В каком случае будет истинна формула $\forall x\, \exists x\, A(x)$? Тот же вопрос для формулы $\exists x\, \forall x\, A(x)$. (Ответ: первая из этих формул эквивалентна формуле $\exists x\, A(x)$, а вторая — формуле $\forall x\, A(x)$.)*

Формула называется *замкнутой*, если она не имеет параметров. Замкнутые формулы называют также *суждениями*. Как мы доказали, истинность замкнутой формулы определяется выбором интерпретации (и не зависит от значений переменных).

§3. Выразимые предикаты

Пусть фиксирована некоторая сигнатура σ и её интерпретация с носителем M. Мы хотим определить понятие *выразимого* (с помощью формулы данной сигнатуры в данной интерпретации) k-местного предиката.

Выберем k переменных $x_1, \ldots, x_k$. Рассмотрим произвольную формулу φ, все параметры которой содержатся в списке $x_1, \ldots, x_k$. Истинность этой формулы зависит только от значений переменных $x_1, \ldots, x_k$. Тем самым возникает отображение $M^k \to \mathbb{B} = \{\mathbf{И}, \mathbf{Л}\}$, то есть некоторый k-местный предикат на M. Говорят, что этот предикат *выражается* формулой φ. Все предикаты, которые можно получить таким способом, называются *выразимыми* (Ясно, что конкретный выбор списка переменных роли не играет.)

Соответствующие им подмножества множества M^k (области истинности выразимых предикатов) также называют выразимыми.

Задача 123. *Докажите, что пересечение, объединение и разность двух выразимых множеств являются выразимыми. Докажите, что проекция k-мерного выразимого множества вдоль одной из "осей координат" является $(k-1)$-мерным выразимым множеством.*

Пример. Сигнатура содержит одноместный функциональный символ S и двуместный предикатный символ равенства ($=$). Рассмотрим интерпретацию этой сигнатуры. В качестве носителя выберем натуральный ряд $\mathbb{N}$. Символ S будет обозначать функцию прибавления единицы (можно считать S сокращением от слова successor — последователь). Знак равенства интерпретируется как совпадение элементов.

Легко проверить, что одноместный предикат "быть нулём" выразим в этой интерпретации, несмотря на то, что константы для нуля в сигнатуре не предусмотрено. В самом деле, он выражается формулой

$$\neg\exists y(x = S(y))$$

с единственным параметром x.

Ещё проще выразить в этой сигнатуре двуместный предикат "быть больше на 2", при этом даже не нужны кванторы: $y = S(S(x))$.

Любопытно, что уже в такой простой ситуации можно сформулировать содержательную задачу: выразить предикат $y = x + N$, где N — большое число (скажем, миллиард), с помощью существенно более короткой формулы, чем $y = S(S(\dots(S(x))\dots))$. Как ни удивительно, это вполне возможно, и соответствующую формулу вполне можно уместить на листе бумаги.

Задача 124. *Докажите, что предикат $y = x + N$ можно выразить формулой указанной сигнатуры, длина которой есть $O(\log N)$. (Указание. Если мы научились выражать $y = x + n$, можно выразить $y = x + 2n$ с помощью формулы*

$$\exists z\,((z = x + n) \wedge (y = z + n))$$

(в которой через $z = x + n$ и $y = z + n$ обозначены соответствующие формулы). Это само по себе ничего не даёт, так как длина формулы увеличилась вдвое, но можно использовать такой трюк:

$$\exists z\,\forall u\,\forall v(((u = x \wedge v = z) \vee (u = z \wedge v = y)) \to (v = u + n)).$$

Далее можно воспользоваться записью числа N в двоичной системе счисления.)

Можно доказать, что в этой сигнатуре кванторы почти не увеличивают набор выразимых предикатов: всякий выразимый предикат будет выражаться бескванторной формулой (возможно, гораздо более длинной), если добавить к сигнатуре константу 0.

Чтобы привыкнуть к понятию выразимости, рассмотрим ещё один пример. Пусть сигнатура содержит предикат равенства и трёхместный предикат C. Рассмотрим интерпретацию, в которой носителем является множество точек плоскости, равенство интерпретируется как совпадение точек, а $C(x, y, z)$ означает, что точки x и y равноудалены от точки z. Оказывается, что этого предиката достаточно, чтобы выразить более или менее все традиционные понятия элементарной геометрии.

Как, например, записать, что три различные точки A, B, C лежат на одной прямой? Вот как: "не существует другой точки C', которая находилась бы на тех же расстояниях от A и B, что и точка C".

Задача 125. *Напишите соответствующую формулу указанной сигнатуры.*

Теперь легко выразить такое свойство четырёх точек A, B, C, D: "точки A и B различны, точки C и D различны и прямые AB и CD параллельны". В самом деле, надо написать, что нет точки, которая бы одновременно лежала на одной прямой с A и B, а также на одной прямой с C и D.

После этого можно выразить свойство четырёх точек "быть вершинами параллелограмма". Это позволяет переносить отрезок параллельно себе. После этого несложно выразить такое свойство: "расстояние AB равно расстоянию CD".

Задача 126. *Запишите соответствующую формулу.*

Аналогичным образом можно двигаться и дальше.

Задача 127. *Выразите свойство $|OA| \leqslant |OB|$ трёх точек O, A, B. (Указание. Напишите, что все прямые, проходящие через A, пересекаются с окружностью радиуса OB с центром в O.)*

Задача 128. *Запишите в виде формулы: (**а**) равенство треугольников; (**б**) равенство углов; (**в**) свойство угла быть прямым.*

Задача 129. *Рассмотрим естественную интерпретацию сигнатуры $(=, <)$ на множестве целых чисел. Как выразить предикат $y = x + 1$?*

Задача 130. *Рассмотрим множество действительных чисел как интерпретацию сигнатуры $(=, +, y = x^2)$. Как выразить трёхместный предикат $xy = z$?*

Задача 131. *Рассмотрим множество целых положительных чисел как интерпретацию сигнатуры, содержащей равенство и двуместный предикат "x делит y". Выразить свойства "равняться единице" и "быть простым числом".*

Задача 132. *Рассмотрим плоскость как интерпретацию сигнатуры, содержащей предикат равенства (совпадение точек) и двуместный предикат "находиться на расстоянии 1". Выразить двуместные предикаты "находиться на расстоянии 2" и "находиться на расстоянии не более 2".*

§4. Выразимость в арифметике

Рассмотрим сигнатуру, имеющую два двуместных функциональных символа — сложение и умножение (как обычно, мы будем писать $x + y$ вместо $+(x, y)$ и т. д.) и двуместный предикатный символ равенства. Рассмотрим интерпретацию этой сигнатуры, носителем которой является множество $\mathbb{N}$ натуральных чисел, а сложение, умножение и равенство интерпретируются стандартным образом.

Выразимые с помощью формул этой сигнатуры предикаты называются *арифметическими* и играют в математической логике важную роль. Соответствующие множества также называются *арифметическими*. О них подробно рассказано в другой главе; оказывается, что почти всякое множество, которое можно описать словами, является арифметическим.

Задача 133. *Докажите, что существует множество натуральных чисел, не являющееся арифметическим. (Указание: семейство всех подмножеств множества $\mathbb{N}$ несчётно, а арифметических множеств счётное число.)*

Для начала мы установим арифметичность довольно простых предикатов.

- Предикат $x \leqslant y$ является арифметическим. В самом деле, его можно записать как $\exists z\,(x + z = y)$.
- Предикаты $x = 0$ и $x = 1$ являются арифметическими. В самом деле, $x = 0$ тогда и только тогда, когда $x \leqslant y$ для любого y (а также когда $x + x = x$). А $x = 1$ тогда и только тогда, когда x представляет собой наименьшее число, отличное от нуля. (Можно также воспользоваться тем, что $y \cdot 1 = y$ при любом y.)

- Вообще для любого фиксированного числа c предикат $x = c$ является арифметическим. (Например, можно написать сумму из большого числа единиц.)
- Полезно такое общее наблюдение: если мы уже установили, что какой-то предикат является арифметическим, то в дальнейшей его можно использовать в формулах, как если бы он входил в сигнатуру, поскольку его всегда можно заменить на выражающую его формулу.
- Предикат $x|y$ (число x является делителем числа y), очевидно, арифметичен (формула $\exists z\,(xz = y)$).
- Предикат "x — простое число" арифметичен. В самом деле, число просто, если оно отлично от 1 и любой его делитель равен 1 или самому числу. Это сразу же записывается в виде формулы.
- Операции частного и остатка при делении арифметичны (в том смысле, что трёхместные предикаты "q есть частное при делении a на b" и "r есть остаток при делении a на b" арифметичны. Например, первый из них записывается формулой $\exists r\,((a = bq + r) \wedge (r < b))$ (как мы уже говорили, использование арифметического предиката $(r < b)$ не создаёт проблем).
- Этот список можно продолжать: для многих предикатов их определение по существу уже является нужной формулой. Например, свойства "быть наибольшим общим делителем", "быть наименьшим общим кратным", "быть взаимно простыми" все относятся к этой категории.
- Предикат "быть степенью двойки" является арифметическим (хотя это и не столь очевидно, как в предыдущих примерах). В самом деле, это свойство можно переформулировать так: любой делитель либо равен единице, либо чётен.

Последнее из наших рассуждений годится для степеней тройки и вообще для степеней любого простого числа. Однако уже для степеней четвёрки оно не проходит, и, пожалуй, мы подошли к границе, где без некоторого общего метода не обойтись.

Два наиболее известных способа доказывать арифметичность основаны на возможности "кодирования" конечных множеств и последовательностей. Один восходит к Гёделю (так называемая β-функция Гёделя), второй изложен в книге "Теория формальных систем. Её написал Р. Смаллиан, известный также как автор популярных сборников "логических задач" и анекдотов. (Один из таких сборников имеет парадоксальное название "Как же называется эта книга?).

В некоторых отношениях метод Гёделя предпочтительней, и мы расска-

зываем о нём ниже, но сейчас для разнообразия рассмотрим другой способ. Зафиксируем взаимно однозначное соответствие между натуральными числами и двоичными словами:

0	1	2	3	4	5	6	7	8	...
Λ	0	1	00	01	10	11	000	001	...

Это соответствие задаётся так: чтобы получить слово, соответствующее числу n, надо записать $n+1$ в двоичной системе и удалить первую единицу. Например, нулю соответствует пустое слово Λ, числу 15 — слово 0000 и т. д. Теперь можно говорить об арифметичности предикатов, определённых на двоичных словах, имея в виду арифметичность соответствующих предикатов на $\mathbb{N}$.

- Предикат "слово x состоит из одних нулей" арифметичен. В самом деле, при переходе к числам ему соответствует предикат $x+1$ есть степень двойки", который (как мы видели) арифметичен.
- Предикат "слова x и y имеют одинаковую длину" арифметичен. В самом деле, это означает, что найдётся степень двойки c, для которой $c-1 \leqslant x, y < 2c-1$ (именно такой промежуток заполняют числа, которым соответствуют слова одной длины).
- Предикат "слово z является конкатенацией слов x и y" (проще говоря, z получается приписыванием y справа к слову x) арифметичен. В самом деле, его можно выразить так: найдётся слово y' из одних нулей, имеющее ту же длину, что и слово y, при этом $(z+1) = (x+1)(y'+1) + (y-y')$ (умножение на $y'+1$ соответствует дописыванию нулей, а добавление $y-y'$ заменяет нули на буквы слова y).
- Предикат "слово x является началом слова y" арифметичен. В самом деле, это означает, что существует слово t, при котором y есть конкатенация x и t.
- То же самое верно для предикатов "x есть конец слова y", "x есть подслово слова y" (последнее означает, что найдутся слова u и v, для которых y есть конкатенация u, x и v; конкатенация трёх слов выразима через конкатенацию двух).
- Существует арифметический трёхместный предикат $S(x,a,b)$ с такими свойствами: (**а**) для любых a и b множество $S_{ab} = \{x \mid S(x,a,b)\}$ конечно; (**б**) среди множеств S_{ab} при различных парах a, b встречаются все конечные множества. Например, в качестве такого предиката можно взять "axa есть подслово слова b" (здесь axa есть конкатенация трёх слов: a, x и снова a).

В самом деле, ясно, что слово x не длиннее слова b, и потому мно-

жество S_{ab} всегда конечно. С другой стороны, пусть имеется некоторое конечное множество слов $x_1, \ldots, x_n$. Положим $a = 100\ldots001$, где число нулей больше длины любого из слов x_i, и $b = ax_1ax_2a\ldots ax_na$.

Последнее утверждение не упоминает явно о словах, и больше они нам не понадобятся: достаточно знать, что конечные множества натуральных чисел можно кодировать парами натуральных чисел в описанном смысле.

Теперь мы можем выразить, что число x является степенью числа 4, следующим образом: существует конечное множество U, которое содержит число x и обладает таким свойством: всякий элемент $u \in U$ либо равен 1, либо делится на 4 и $u/4$ также принадлежит U. Теперь надо везде заменить множество U на его код u_1, u_2, а утверждение $x \in U$ на $S(x, u_1, u_2)$, где S — построенный нами кодирующий предикат.

Немного сложнее выразить двуместный предикат $x = 4^k$. Здесь нам хотелось бы сказать так: существует последовательность $x_0, x_1, \ldots, x_k$, для которой $x_0 = 1$, каждый следующий член вчетверо больше предыдущего ($x_{i+1} = 4x_i$) и $x_k = x$. Как научиться говорить о последовательностях, если мы умеем говорить о множествах? Вспомним, что в терминах теории множеств последовательность есть функция, определённая на начальном отрезке натурального ряда, то есть конечное множество пар $\{\langle 0, x_0\rangle, \langle 1, x_1\rangle, \ldots, \langle k, x_k\rangle\}$. Пары можно кодировать числами. Например, можно считать кодом пары $\langle x, y\rangle$ число $c = (x+y)^2 + x$, поскольку по нему арифметически восстанавливается $x + y$ (как наибольшее число, квадрат которого не превосходит c), а затем x и y. Теперь конечное множество пар можно заменить конечным множеством их кодов, которое в свою очередь можно закодировать парой чисел.

Задача 134. *Проведите это рассуждение подробно.*

Задача 135. *Покажите, что двуместный предикат "x есть n-ое по порядку простое число" арифметичен.*

§5. Невыразимые предикаты: автоморфизмы

Мы видели, как можно доказать выразимость некоторых свойств. Сейчас мы покажем, каким образом можно доказывать невыразимость.

Начнём с такого примера. Пусть сигнатура содержит двуместный предикат равенства ($=$) и двуместную операцию сложения ($+$). Рассмотрим её интерпретацию, носителем которой являются целые числа, а равенство и сложение интерпретируются стандартным образом. Оказывается, что предикат $x > y$ не является выразимым.

Причина очевидна: с точки зрения сложения целые числа устроены симметрично, положительные ничем не отличаются от отрицательных. Если мы изменим знак у всех переменных, входящих в формулу, то её истинность не может измениться. Но при этом $x > y$ заменится на $x < y$, и потому это свойство не является выразимым.

Формально говоря, надо доказывать по индукции такое свойство: если формула φ указанной сигнатуры истинна при оценке π, то она истинна и при оценке π', в которой значения всех переменных меняют знак. (Подробно мы объясним это в общей ситуации дальше.)

Сформулируем общую схему, которой следует это рассуждение. Пусть имеется некоторая сигнатура σ и интерпретация этой сигнатуры, носителем которой является множество M. Взаимно однозначное отображение $\alpha\colon M \to M$ называется *автоморфизмом* интерпретации, если все функции и предикаты, входящие в интерпретацию, устойчивы относительно α. При этом k-местный предикат P называется *устойчивым* относительно α, если

$$P(\alpha(m_1), \ldots, \alpha(m_k)) \Leftrightarrow P(m_1, \ldots, m_k)$$

для любых элементов $m_1, \ldots, m_k \in M$. Аналогичным образом k-местная функция f называется *устойчивой* относительно α, если

$$f(\alpha(m_1), \ldots, \alpha(m_k)) = \alpha(f(m_1, \ldots, m_k)).$$

Это определение обобщает стандартное определение автоморфизма для групп, колец, полей и т. д.

Теорема 36. *Предикат, выразимый в данной интерпретации, устойчив относительно её автоморфизмов.*

Доказательство. Проведём доказательство этого (достаточно очевидного) утверждения формально.

Пусть π — некоторая оценка, то есть отображение, ставящее в соответствие всем индивидным переменным некоторые элементы носителя. Через $\alpha \circ \pi$ обозначим оценку, которая получится, если к значению каждой переменной применить отображение α; другими словами, $\alpha \circ \pi(\xi) = \alpha(\pi(\xi))$ для любой переменной ξ.

Первый шаг состоит в том, чтобы индукцией по построению терма t доказать такое утверждение: значение терма t при оценке $\alpha \circ \pi$ получается применением α к значению терма t при оценке π:

$$[t](\alpha \circ \pi) = \alpha([t](\pi)).$$

Для переменных это очевидно, а шаг индукции использует устойчивость всех функций интерпретации относительно α.

Теперь индукцией по построению формулы φ легко доказать такое утверждение:

$$[\varphi](\alpha \circ \pi) = [\varphi](\pi).$$

Мы не будем выписывать эту проверку; скажем лишь, что взаимная однозначность α используется, когда мы разбираем случай кванторов. (В самом деле, если с одной стороны изоморфизма берётся какой-то объект, то взаимная однозначность позволяет взять соответствующий ему объект с другой стороны изоморфизма.) □

Теорема 36 позволяет доказать невыразимость какого-то предиката, предъявив автоморфизм интерпретации, относительно которого интересующий нас предикат неустойчив. Вот несколько примеров:

- $(\mathbb{Z}, =, <)$ Сигнатура содержит равенство и отношение порядка. Интерпретация: целые числа. Невыразимый предикат: $x = 0$. Автоморфизм: $x \mapsto x + 1$.
- $(\mathbb{Q}, =, <, +)$ Сигнатура содержит равенство, отношение порядка и операцию сложения. Интерпретация: рациональные числа. Невыразимый предикат: $x = 1$. Автоморфизм: $x \mapsto 2x$.

  Заметим, что сложение позволяет выразить предикат $x = 0$. Кроме того, отметим, что вместо рациональных чисел можно взять действительные (но не целые, так как в этом случае единица описывается как наименьшее число, большее нуля).
- $(\mathbb{R}, =, <, 0, 1)$ Сигнатура содержит равенство, порядок и константы 0 и 1. Интерпретация: действительные числа. Невыразимый предикат: $x = 1/2$. (Автоморфизм упорядоченного множества $\mathbb{R}$, сохраняющий 0 и 1, но не $1/2$, построить легко.)
- $(\mathbb{R}, =, +, 0, 1)$ Сигнатура содержит равенство, сложение, константы 0 и 1. Интерпретация: действительные числа. Одноместный предикат $x = \gamma$ выразим для рациональных γ и невыразим для иррациональных γ.

  В самом деле, выразимость для рациональных γ очевидна. Невыразимость для иррациональных γ следует из того, что для любых двух иррациональных γ_1 и γ_2 существует автоморфизм, переводящий γ_1 в γ_2. (В самом деле, рассмотрим $\mathbb{R}$ как бесконечномерное векторное пространство над $\mathbb{Q}$. Векторы $1, \gamma_1$ линейно независимы и потому их можно дополнить до базиса Гамеля (подробности смотри в книжке по теории множеств [**1**]). Сделаем то же самое с векторами $1, \gamma_2$. Получатся равномощные базисы, после чего мы берём $\mathbb{Q}$-линейный оператор, переводящий 1 в 1 и γ_1 в γ_2.)

- $(\mathbb{C}, =, +, \times, 0, 1)$ В сигнатуру входят предикат равенства, операции сложения и умножения, а также константы 0 и 1. Интерпретация: комплексные числа. Предикат $x = \gamma$, где γ — некоторое комплексное число, выразим для рациональных γ и невыразим для иррациональных γ.

  В самом деле, если γ иррационально, то оно может быть алгебраическим или трансцендентным. В первом случае рассмотрим многочлен из $\mathbb{Q}[x]$ минимальной степени, обращающийся в 0 в точке γ; по предположению он имеет степень больше 1 и потому имеет другой корень γ'. В алгебре доказывается, что существует автоморфизм $\mathbb{C}$ над $\mathbb{Q}$, переводящий γ в γ'.

  В случае трансцендентного γ мы используем такой факт: для любых трансцендентных $\gamma_1, \gamma_2 \in \mathbb{C}$ существует автоморфизм поля $\mathbb{C}$ над $\mathbb{Q}$, который переводит γ_1 в γ_2.

  Отметим, что для поля $\mathbb{R}$ вместо $\mathbb{C}$ такое рассуждение не проходит, так как это поле не имеет нетривиальных автоморфизмов. (Отношение порядка выразимо: положительные числа суть квадраты, поэтому любой автоморфизм сохраняет порядок. Поскольку автоморфизм оставляет на месте все рациональные числа, он должен быть тождественным.)

  (В этом случае предикат $x = \gamma$ выразим тогда и только тогда, когда γ — алгебраическое число.)

Задача 136. *Покажите, что предикат $y = x + 1$ невыразим в интерпретации $(\mathbb{Z}, =, f)$, где f — одноместная функция $x \mapsto (x + 2)$.*

Задача 137. *Покажите, что предикат $x = 2$ невыразим в множестве целых положительных чисел с предикатами равенства и "x делит y".*

ГЛАВА VI

Исчисление предикатов

§1. Общезначимые формулы

Исчисление высказываний (глава IV) позволяло выводить все тавтологии из некоторого набора базисных тавтологий (названных аксиомами) с помощью некоторых правил вывода (на самом деле единственного правила modus ponens). Сейчас мы хотим решить аналогичную задачу для формул первого порядка. Соответствующее исчисление называется *исчислением предикатов*.

Пусть фиксирована некоторая сигнатура σ. Формула φ этой сигнатуры (возможно, с параметрами) называется *общезначимой*, если она истинна в любой интерпретации сигнатуры σ на любой оценке.

Общезначимые формулы в логике предикатов играют ту же роль, что тавтологии в логике высказываний. Между ними есть и формальная связь: если взять любую тавтологию и вместо входящих в неё пропозициональных переменных подставить произвольные формулы сигнатуры σ, получится общезначимая формула. В самом деле, пусть есть некоторая интерпретация сигнатуры σ и некоторая оценка (то есть фиксированы значения индивидных переменных). Тогда каждая из подставленных формул станет истинной или ложной, а значение всей формулы определяется с помощью таблиц истинности для логических связок, то есть по тем же правилам, что в логике высказываний.

Конечно, бывают и другие общезначимые формулы, не являющиеся частным случаями пропозициональных тавтологий. Например, формула

$$\forall x\, A(x) \to \exists y A(y)$$

общезначима (здесь существенно, что носитель любой интерпретации непуст). Другие примеры общезначимых формул (во втором случае φ — произвольная формула):

$$\exists y\, \forall x\, B(x,y) \to \forall x\, \exists y\, B(x,y), \qquad \neg\forall x\, \neg\varphi \to \exists x\, \varphi.$$

Задача 138. *Будет ли общезначима формула*
*(***а***)* $\forall x\, \exists y\, B(x,y) \to \exists y\, \forall x\, B(x,y)$*;*
*(***б***)* $\neg\forall x\, \exists y\, B(x,y) \to \exists x\, \forall y\, \neg B(x,y)$*?*

Многие вопросы можно сформулировать как вопросы об общезначимости некоторых формул. Например, можно записать свойства рефлексивности, транзитивности и антисимметричности в виде формул R, T и A сигнатуры $(=, <)$ и затем написать формулу

$$R \wedge T \wedge A \to \exists x \, \forall y \, ((y < x) \vee (y = x)).$$

Общезначимость этой формулы означала бы, что любое линейно упорядоченное множество имеет наибольший элемент, так что она не общезначима.

Задача 139. *Напишите формулы R, T, A и проверьте, что приведённая нами формула не общезначима, хотя истинна во всех конечных интерпретациях.*

Две формулы φ и ψ (с параметрами или без) называются *эквивалентными*, если в любой интерпретации и на любой оценке, на которой истинна одна из них, истинна и другая. Это определение равносильно такому: формула $\varphi \leftrightarrow \psi$ общезначима. Здесь, напомним, $\varphi \leftrightarrow \psi$ есть сокращение для $((\varphi \to \psi) \wedge (\psi \to \varphi))$.

Общезначимость любой формулы φ очевидно равносильна общезначимости её *замыкания* — формулы, которая получится, если слева к φ приписать кванторы всеобщности по всем параметрам.

Двойственное к общезначимости понятие — выполнимость. Формула называется *выполнимой*, если она истинна в некоторой интерпретации на некоторой оценке. Очевидно, формула φ общезначима тогда и только тогда, когда формула $\neg\varphi$ не является выполнимой.

Задача 140. *Закончите утверждение: выполнимость формулы с параметрами равносильна выполнимости замкнутой формулы, которая получится, если...*

Чтобы проверить, является ли формула тавтологией, достаточно подставить в неё все возможные наборы значений переменных. Хотя этот процесс может быть на практике невыполним (наборов слишком много), теоретически мы имеем простой алгоритм проверки, является ли формула тавтологией. Для общезначимых формул в общем случае такого алгоритма не существует (теорема Чёрча; её доказательство можно прочесть в [**3**]); он есть только для очень ограниченных классов формул. Например, если сигнатура содержит только нульместные предикатные символы, то задача по существу сводится к проверке тавтологичности (в этом случае кванторы фиктивны). Чуть более сложен случай с одноместными предикатами.

Задача 141. *Пусть сигнатура σ содержит только одноместные предикаты. Докажите, что всякая выполнимая формула этой сигнатуры, содержащая n различных предикатов, выполнима в некоторой конечной*

интерпретации, содержащей не более 2^n элементов. Как использовать этот факт для алгоритмической проверки выполнимости формул такой сигнатуры?

§2. Аксиомы и правила вывода

Возвратимся к нашей задаче: какие аксиомы и правила вывода нам нужны, чтобы получить все общезначимые формулы некоторой сигнатуры σ? Естественно использовать все схемы аксиом (1) – (11) исчисления высказываний (раздел 1), но только вместо букв A, B и C теперь можно подставлять произвольные формулы сигнатуры σ. Теорема о полноте исчисления высказываний гарантирует, что после этого мы сможем вывести любой частный случай любой пропозициональной тавтологии (то есть любую формулу, которая получается из пропозициональной тавтологии заменой пропозициональных переменных на формулы сигнатуры σ). В самом деле, возьмём вывод этой тавтологии в исчислении высказываний (которое, как мы знаем, полно) и выполним соответствующую замену во всех формулах этого вывода.

Почти столь же просто понять, что ничего другого такие аксиомы не дадут: если пользоваться лишь схемами аксиом (1) – (11), разрешая брать в них в качестве A, B, C произвольные формулы сигнатуры σ, а в качестве правила вывода использовать modus ponens, то все выводимые формулы будут частными случаями пропозициональных тавтологий. В самом деле, если какая-то подформула начинается с квантора, то в выводе она может встречаться только как единое целое, то есть такая подформула ведёт себя как пропозициональная переменная.

Задача 142. *Проведите это рассуждение аккуратно.*

Это наблюдение скорее тривиально, чем удивительно — если среди наших аксиом и правил вывода нет ничего о смысле кванторов, то формулы, начинающиеся с кванторов, будут вести себя как неделимые блоки. Таким образом, нам нужны аксиомы и правила вывода, отражающие интуитивный смысл кванторов.

Вспомним, как выглядели аксиомы исчисления высказываний. У нас было два типа аксиом для конъюнкции и дизъюнкции: одни говорили, что из них следует (например, из $A \wedge B$ следовало B), а другие — как их можно доказать (например, аксиома $(A \to (B \to (A \wedge B)))$ говорила, что для доказательства $(A \wedge B)$ надо доказать A и B). Кванторы всеобщности и существования в некотором смысле аналогичны конъюнкции и дизъюнкции, и аксиомы для них тоже будут похожими. Например, среди аксиом будет

формула

$$\forall x\, A(x) \to A(t),$$

где A — одноместный предикатный символ нашей сигнатуры, а t — константа, переменная или вообще любой терм. (Если A верно для всех x, то оно должно быть верно и для нашего конкретного t. Можно сказать и так: из "бесконечной конъюнкции" всех $A(x)$ вытекает один из её членов.)

Конечно, такую аксиому надо иметь не только для одноместного предикатного символа A, но для любой формулы φ, любой переменной ξ и любого терма t. Естественно сказать, что если φ — любая формула, а t — любой терм, то формула

$$\forall \xi\, \varphi \to \varphi(t/\xi),$$

где $\varphi(t/\xi)$ обозначает результат подстановки t вместо всех вхождений переменной ξ в формулу φ, является аксиомой. (Запись $\varphi(t/\xi)$ можно читать как "фи от тэ вместо кси".)

К сожалению, всё не так просто. Например, если формула φ имеет вид

$$A(x) \wedge \exists x\, B(x, x),$$

то подстановка терма $f(y)$ вместо x даст абсурдное выражение

$$A(f(y)) \wedge \exists f(y)\, B(f(y), f(y)),$$

вообще не являющееся формулой. А если подставить $f(y)$ только внутри A и B, то получится выражение

$$A(f(y)) \wedge \exists x\, B(f(y), f(y)),$$

которое хотя и будет формулой, но имеет совсем не тот смысл, который нам нужен.

Конечно, в данном случае по смыслу ясно, что подставлять $f(y)$ надо лишь вместо самого первого вхождения переменной x. Но если мы хотим определить формальную систему аксиом и правил вывода, то надо дать формальные определения.

Для каждого квантора в формуле рассмотрим его *область действия* — начинающуюся с него подформулу. *Свободным вхождением* индивидной переменной в формулу называется вхождение, не попадающее в область действия одноимённого квантора. Легко понять, что это определение можно переформулировать индуктивно:

- любое вхождение переменной в терм свободно;
- свободные вхождения переменной в формулу φ являются её свободными вхождениями в формулу $\neg\varphi$;
- свободные вхождения любой переменной в одну из формул φ и ψ являются свободными вхождениями в $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ и $(\varphi \to \psi)$;

- переменная ξ не имеет свободных вхождений в формулы $\forall\xi\,\varphi$ и $\exists\xi\,\varphi$; свободные вхождения остальных переменных в φ являются свободными вхождениями в эти две формулы.

Сравнивая это определение с индуктивным определением параметров формулы в разделе 2, мы видим, что параметры — это переменные, имеющие свободные вхождения в формулу.

Вхождения переменной, не являющиеся свободными (в том числе стоящие рядом с квантором) называют *связанными*. Например, переменная x имеет одно свободное и три связанных вхождения в формулу $A(x) \wedge \exists x\, B(x,x)$.

Теперь можно внести поправку в сказанное выше и считать, что аксиомами являются формулы

$$\forall\xi\,\varphi \to \varphi(t/\xi),$$

где $\varphi(t/\xi)$ есть результат подстановки t вместо всех *свободных* вхождений переменной ξ. Однако такой оговорки недостаточно, как показывает следующий пример.

Подставляя $f(y)$ вместо x в формулу $\forall z\, B(x,z)$, мы получаем (в полном согласии с нашей интуицией) формулу $\forall z\, B(f(y),z)$. Теперь рассмотрим формулу $\forall y\, B(x,y)$, которая отличается от $\forall z\, B(x,z)$ лишь именем связанной переменной и должна иметь тот же смысл. Переменная x в ней по-прежнему свободна, но подстановка $f(y)$ вместо x даёт формулу $\forall y\, B(f(y),y))$, в которой $f(y)$ неожиданно для себя попадает в область действия квантора по y. Такое явление иногда называют *коллизией переменных*; при этом подстановка даёт формулу, имеющую совсем не тот смысл, какой мы хотели.

Задача 143. *Приведите пример формулы вида $\forall\xi\,\varphi \to \varphi(t/\xi)$, в которой происходит коллизия переменных и которая не является общезначимой. (Ответ: $\forall x\,\exists y\, A(x,y) \to \exists y A(y,y)$.)*

Поэтому нам придётся принять ещё одну меру предосторожности и формально определить понятие *корректной* подстановки терма вместо переменной. Мы будем говорить, что подстановка терма t вместо переменной ξ корректна, если в процессе текстуальной замены всех свободных вхождений переменной ξ на терм t никакая переменная из t не попадает в область действия одноимённого квантора.

Педантичный читатель мог бы попросить доказать, что результат такой подстановки будет формулой. Это проще всего сделать так: дать индуктивное определение корректной подстановки, равносильное исходному.

Сначала определим индуктивно результат подстановки терма t вместо переменной ξ в терм u; этот результат будем обозначать $u(t/\xi)$:

- $\xi(t/\xi)$ есть t; для любой переменной η, отличной от ξ, мы полагаем $\eta(t/\xi)$ равным η.
- если f есть k-местный функциональный символ, а $t_1, \ldots, t_k$ — термы, то
$$f(t_1, \ldots, t_k)(t/\xi) = f(t_1(t/\xi), \ldots, t_k(t/\xi)).$$

Теперь индуктивное определение продолжается для формул:

- для атомарных формул: если R есть k-местный предикатный символ, а $t_1, \ldots, t_k$ — термы, то
$$R(t_1, \ldots, t_k)(t/\xi) = R(t_1(t/\xi), \ldots, t_k(t/\xi))$$
и подстановка является корректной;
- подстановка терма t вместо переменной ξ в формулу $\neg\varphi$ корректна, если она корректна для формулы φ, при этом
$$[\neg\varphi](t/\xi) = \neg[\varphi(t/\xi)]$$
(квадратные скобки указывают порядок действий, не являясь частью формулы);
- подстановка терма t вместо переменной ξ в формулу $(\varphi \wedge \psi)$ корректна, если она корректна для обеих формул φ и ψ, при этом
$$(\varphi \wedge \psi)(t/\xi) = (\varphi(t/\xi) \wedge \psi(t/\xi));$$
аналогично для формул $(\varphi \vee \psi)$ и $(\varphi \to \psi)$;
- наконец, подстановка t вместо ξ в формулу $\forall\eta\,\varphi$ корректна в двух случаях:

  (1) если ξ не является параметром формулы $\forall\eta\,\varphi$ (это возможно, когда ξ не является параметром φ или когда ξ совпадает с η); при этом подстановка ничего не меняет в формуле;

  (2) переменная ξ является параметром формулы $\forall\eta\,\varphi$, но переменная η не входит в терм t и подстановка $\varphi(t/\xi)$ корректна; при этом
$$[\forall\eta\,\varphi](t/\xi) = \forall\eta\,[\varphi(t/\xi)].$$

Аналогично определяется корректная подстановка в формулу $\exists\xi\varphi$.

Главная часть в этом определении — последний его пункт, который, во-первых, говорит, что вместо связанных вхождений переменных ничего подставлять не надо, а во-вторых, требует, чтобы при корректной подстановке переменные из терма t не подпадали под действие одноимённых кванторов.

После всех этих приготовлений мы можем сформулировать две оставшиеся схемы аксиом исчисления предикатов: формула

(12) $\forall\xi\,\varphi \to \varphi(t/\xi)$

и двойственная ей формула

(13) $\varphi(t/\xi) \to \exists\xi\,\varphi$

будут аксиомами исчисления предикатов, если указанные в них подстановки корректны.

Два частных случая, когда подстановка заведомо корректна: во-первых, можно безопасно подставлять константу (или любой терм без параметров), во-вторых, подстановка переменной вместо себя всегда корректна (и ничего не меняет в формуле).

Отсюда следует, что формулы $\forall\xi\,\varphi \to \varphi$ и $\varphi \to \exists\xi\,\varphi$ будут аксиомами исчисления предикатов (для любой формулы φ и переменной ξ).

Нужны ли нам ещё какие-нибудь аксиомы и правила вывода? Конечно, нужны, поскольку уже сформулированные аксиомы не полностью отражают смысл кванторов. (Например, они вполне согласуются с таким пониманием этого смысла: формула $\forall\xi\,\varphi$ всегда ложна, а формула $\exists\xi\,\varphi$ всегда истинна.) Поэтому мы введём в наше исчисление два правила вывода, называемые *правилами Бернайса*, и на этом определение исчисления предикатов будет завершено.

Если переменная ξ не является параметром формулы ψ, то правила Бернайса разрешают такие переходы:

$$\frac{\psi \to \varphi}{\psi \to \forall\xi\,\varphi} \qquad \frac{\varphi \to \psi}{\exists\xi\,\varphi \to \psi}$$

Мы говорим, что стоящая снизу от черты (в каждом из правил) формула получается по соответствующему правилу из верхней. Соответственно дополняется и определение вывода как последовательности формул, в которой каждая формула либо является аксиомой, либо получается из предыдущих по одному из правил вывода (раньше было только правило MP, теперь добавились два новых правила).

Поясним интуитивный смысл этих правил. Первое говорит, что если из ψ следует φ, причём в φ есть параметр ξ, которого нет в ψ, то это означает, что формула φ истинна при всех значениях параметра ξ, если только формула ψ истинна.

Используя первое правило Бернайса, легко установить допустимость *правила обобщения*

$$\frac{\varphi}{\forall\xi\,\varphi} \qquad \text{(Gen)}$$

(если в исчислении предикатов выводима формула сверху от черты, то выводима и формула снизу). В самом деле, возьмём какую-нибудь выводимую формулу ψ без параметров (например, аксиому, в которой вместо A, B и C

подставлены замкнутые формулы). Раз выводима формула φ, то выводима и формула $\psi \to \varphi$ (поскольку $\varphi \to (\psi \to \varphi)$ является тавтологией и даже аксиомой). Теперь по правилу Бернайса выводим $\psi \to \forall\xi\,\varphi$ и применяем правило MP к этой формуле и к формуле ψ.

Правило (Gen) (от Generalization — обобщение) кодифицирует стандартную практику рассуждений: мы доказываем какое-то утверждение φ со свободной переменной ξ, после чего заключаем, что мы доказали $\forall\xi\,\varphi$, так как ξ было произвольным.

Второе правило Бернайса также вполне естественно: желая доказать ψ в предположении $\exists\xi\,\varphi$, мы говорим: пусть такое ξ существует, возьмём его и докажем ψ (то есть докажем $\varphi \to \psi$ со свободной переменной ξ).

Задача 144. *Покажите, что класс выводимых в исчислении предикатов формул не изменится, если мы вместо правил Бернайса добавим туда правило обобщения и две аксиомы*

$$\forall\xi\,(\psi \to \varphi) \to (\psi \to \forall\xi\,\varphi)$$

и

$$\forall\xi\,(\varphi \to \psi) \to (\exists\xi\,\varphi \to \psi)$$

(в которых требуется, чтобы переменная ξ не была параметром формулы ψ).

Как и в случае исчисления высказываний, перед нами стоят две задачи: надо доказать корректность исчисления предикатов (всякая выводимая формула общезначима) и его полноту (всякая общезначимая формула выводима). Этим мы и займёмся в следующих разделах.

§3. Корректность исчисления предикатов

Теорема 37. *Всякая выводимая в исчислении предикатов формула является общезначимой.*

Доказательство. Для исчисления высказываний проверка корректности была тривиальной — надо было по таблице проверить, что все аксиомы (1) – (11) являются тавтологиями. С этими аксиомами и сейчас нет проблем. Но в двух следующих аксиомах есть ограничение на корректность подстановки, без которого они могут не быть общезначимыми. Естественно, это ограничение должно быть использовано и в доказательстве корректности, и это потребует довольно скучных рассуждений — тем более скучных, что сам факт кажется ясным и так. Тем не менее такие рассуждения надо уметь проводить, так что мы ничего пропускать не будем.

Итак, пусть фиксирована сигнатура σ, а также некоторая интерпретация этой сигнатуры. Всюду далее, говоря о термах и формулах, мы имеем в виду термы и формулы этой сигнатуры, а говоря об их значениях, имеем в виду значения в этой интерпретации.

Лемма 1. Пусть u и t — термы, а ξ — переменная. Тогда

$$[u(t/\xi)](\pi) = [u](\pi + (\xi \mapsto [t](\pi)))$$

для произвольной оценки π.

Напомним обозначения: в левой части мы подставляем t вместо ξ в терм u, и берём значение получившегося терма на оценке π. В правой части стоит значение терма u на оценке, которая получится из π, если значение переменной ξ изменить и считать равным значению терма t на оценке π.

В сущности, это утверждение совершенно тривиально: оно говорит, например, что значение $\sin(\cos(x))$ при $x = 2$ равно значению $\sin(y)$ при $y = \cos(2)$. Но раз уж мы взялись всё доказывать формально, докажем его индукцией по построению терма u. Если терм u есть переменная, отличная от ξ, то ни подстановка, ни изменение оценки не сказываются на значении терма u. Для случая $u = \xi$ получаем $[t](\pi)$ слева и справа. Если терм получается из других термов применением функционального символа, то подстановка выполняется отдельно в каждом из этих термов, так что искомое равенство также сохраняется. Лемма 1 доказана.

Аналогичное утверждение для формул таково:

Лемма 2. Пусть φ — формула, t — терм, а ξ — переменная, причём подстановка t вместо ξ в формулу φ корректна. Тогда

$$[\varphi(t/\xi)](\pi) = [\varphi](\pi + (\xi \mapsto [t](\pi)))$$

для произвольной оценки π.

Поясним смысл этой леммы на примере. Пусть ξ является единственным параметром формулы φ, а c — константа. Тогда формула $\varphi(c/\xi)$ замкнута; лемма утверждает, что её истинность равносильна истинности φ на оценке, при которой значение переменной ξ есть элемент интерпретации, соответствующий константе c.

Доказательство леммы проведём индукцией по построению формулы φ. Для атомарных формул это утверждение является прямым следствием леммы 1. Кроме того, из определения истинностного значения формулы и из определения подстановки ясно, что если утверждение леммы 2 верно для двух формул φ_1 и φ_2, то оно верно для их любой их логической комбинации (конъюнкции, дизъюнкции и импликации); аналогично для отрицания.

Единственный нетривиальный случай — формула, начинающаяся с квантора. Здесь наши определения вступают в игру. Пусть φ имеет вид $\forall\eta\,\psi$. Есть два принципиально разных случая: либо ξ является параметром

формулы φ (т. е. формулы $\forall\eta\,\psi$), либо нет. Во втором случае $\varphi(t/\xi)$ совпадает с φ, а изменение значения переменной ξ в оценке π не влияет на значение формулы φ, так что всё сходится. Осталось разобрать случай, когда ξ является параметром формулы $\forall\eta\,\psi$ (отсюда следует, что ξ не совпадает с η). По определению корректной подстановки, в этом случае переменная η не входит в терм t и подстановка $\psi(t/\xi)$ корректна. Тогда

$$\begin{aligned}[(\forall\eta\,\psi)(t/\xi)](\pi) &= [\forall\eta\,(\psi(t/\xi))](\pi) = \\ &= \wedge_m[\psi(t/\xi)](\pi + (\eta \mapsto m)) = \\ &= \wedge_m[\psi](\pi + (\eta \mapsto m) + (\xi \mapsto [t](\pi + (\eta \mapsto m)))).\end{aligned}$$

Мы воспользовались определением подстановки, определением истинности ($\wedge_m$ означает конъюнкцию по всем элементам из носителя интерпретации) и предположением индукции для формулы ψ. Теперь надо заметить, что переменная η не входит в t по предположению корректности, и потому значение терма t не изменится, если заменить $\pi + (\eta \mapsto m)$ на π. Далее, ξ и η различны, поэтому два изменения в π можно переставить местами. Используя эти соображения, можно продолжить цепочку равенств:

$$\begin{aligned}&= \wedge_m[\psi](\pi + (\xi \mapsto [t](\pi)) + (\eta \mapsto m)) = \\ &= [\forall\eta\,\psi](\pi + (\xi \mapsto [t](\pi))) = \\ &= [\varphi](\pi + (\xi \mapsto [t](\pi))),\end{aligned}$$

что и требовалось. Случай формулы вида $\exists\xi\,\psi$ разбирается аналогично, надо только $\wedge_m$ заменить на $\vee_m$. Лемма 2 доказана.

Теперь уже ясно, почему формула

$$\forall\xi\,\varphi \to \varphi(t/\xi)$$

будет истинна на любой оценке π (если подстановка корректна). В самом деле, если левая часть импликации истинна на π, то φ будет истинна на любой оценке π', которая отличается от π лишь значением переменной ξ. В частности, φ будет истинна и на оценке $\pi + (\xi \mapsto [t](\pi))$, что по только что доказанной лемме 2 означает, что правая часть импликации истинна на π.

Общезначимость формулы

$$\varphi(t/\xi) \to \exists\xi\,\varphi$$

доказывается аналогично.

Нам осталось проверить, что правила вывода сохраняют общезначимость. Для правила MP это очевидно (как и в случае исчисления высказываний). Проверим это для правил Бернайса. Это совсем несложно, так как здесь нет речи ни о каких корректных подстановках.

Пусть, например, формула $\psi \to \varphi$ общезначима и переменная ξ не является параметром формулы ψ. Проверим, что формула $\psi \to \forall\xi\,\varphi$ общезначима, то есть истинна на любой оценке π (в любой интерпретации). В самом деле, пусть ψ истинна на оценке π. Тогда она истинна и на любой оценке π', отличающейся от π только значением переменной ξ (значение переменной ξ не влияет на истинность ψ, так как ξ не является параметром). Значит, и формула φ истинна на любой такой оценке π'. А это в точности означает, что $\forall\xi\,\varphi$ истинна на оценке π, что и требовалось.

Для второго правила Бернайса рассуждение симметрично. Пусть формула $\varphi \to \psi$ общезначима и переменная ξ не является параметром формулы ψ. Покажем, что формула $\exists\xi\,\varphi \to \psi$ общезначима. В самом деле, пусть её левая часть истинна на некоторой оценке π. По определению истинности формулы, начинающейся с квантора существования, это означает, что найдётся оценка π', которая отличается от π только на переменной ξ, для которой $[\varphi](\pi')$ истинно. Тогда и $[\psi](\pi')$ истинно. Но переменная ξ не является параметром формулы ψ, так что $[\psi](\pi') = [\psi](\pi)$. Следовательно, формула ψ истинна на оценке π, что и требовалось доказать. □

§4. Выводы в исчислении предикатов

4.1. Примеры выводимых формул

Прежде чем доказывать теорему Гёделя о полноте исчисления предикатов, мы должны приобрести некоторый опыт построения выводов в этом исчислении.

- Прежде всего отметим, что возможность сослаться на теорему о полноте исчисления высказываний и считать выводимым любой частный случай пропозициональной тавтологии сильно облегчает жизнь. Например, пусть мы вывели две формулы φ и ψ и хотим теперь вывести формулу $(\varphi \wedge \psi)$. Это просто: заметим, что формула $(\varphi \to (\psi \to \to (\varphi \wedge \psi)))$ является частным случаем пропозициональной тавтологии (а на самом деле и аксиомой) и дважды применяем правило MP.
- Другой пример такого же рода: если формула $\varphi \to \psi$ выводима, то выводима и формула $\neg\psi \to \neg\varphi$, поскольку импликация

$$(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$$

является частным случаем пропозициональной тавтологии.

- Ещё один пример: если выводимы формулы $\varphi \to \psi$ и $\psi \to \tau$, то выводима и формула $\varphi \to \tau$, поскольку формула

$$(\varphi \to \psi) \to ((\psi \to \tau) \to (\varphi \to \tau))$$

является частным случаем пропозициональной тавтологии.
- Для произвольной формулы φ выведем формулу

$$\forall x\, \varphi \to \exists x\, \varphi.$$

В самом деле, подстановка переменной вместо себя всегда допустима, поэтому формулы $\forall x\, \varphi \to \varphi$ и $\varphi \to \exists x\, \varphi$ являются аксиомами. Остаётся воспользоваться предыдущим замечанием.
- Для произвольной формулы φ выведем формулу

$$\exists y\, \forall x\, \varphi \to \forall x\, \exists y\, \varphi.$$

Формулы $\forall x\, \varphi \to \varphi$ и $\varphi \to \exists y\, \varphi$ являются аксиомами. С их помощью выводим формулу $\forall x\, \varphi \to \exists y\, \varphi$. Теперь заметим, что левая часть импликации не имеет параметра x, а правая часть не имеет параметра y, так что можно применить два правила Бернайса (в любом порядке) и добавить справа квантор $\forall x$, а слева — квантор $\exists y$.
- Предположим, что формула $\varphi \to \psi$ выводима, а ξ — произвольная переменная. Покажем, что в этом случае выводима формула $\forall \xi\, \varphi \to$ $\to \forall \xi\, \psi$. В самом деле, формула $\forall \xi\, \varphi \to \varphi$ является аксиомой. Далее выводим (с помощью пропозициональных тавтологий и правила MP) формулу $\forall \xi\, \varphi \to \psi$; остаётся воспользоваться правилом Бернайса (левая часть не имеет параметра ξ).
- Аналогичным образом из выводимости формулы $\varphi \to \psi$ следует выводимость формулы $\exists \xi\, \varphi \to \exists \xi\, \psi$, только надо начать с аксиомы $\psi \to \exists \xi\, \psi$, затем получить $\varphi \to \exists \xi\, \psi$, а потом применить правило Бернайса.
- Таким образом, если формулы φ и ψ доказуемо эквивалентны (это значит, что импликации $\varphi \to \psi$ и $\psi \to \varphi$ выводимы), то формулы $\forall \xi\, \varphi$ и $\forall \xi\, \psi$ также доказуемо эквивалентны. (Аналогичное утверждение верно и для формул $\exists \xi\, \varphi$ и $\exists \xi\, \psi$.)

Теперь несложно доказать и и более общий факт: замена подформулы на доказуемо эквивалентную даёт доказуемо эквивалентную формулу.
- Выведем формулу $\forall x\, A(x) \to \forall y\, A(y)$ (здесь A — одноместный предикатный символ). Это несложно: начнём с аксиомы $\forall x\, A(x) \to A(y)$, в ней левая часть не имеет параметра y и потому по правилу Бернайса из неё получается искомая формула. Этот пример показывает,

что связанные переменные можно переименовывать, не меняя смысла формулы

- Выведем формулы, связывающие кванторы всеобщности и существования:

$$\forall\xi\,\varphi \leftrightarrow \neg\exists\xi\,\neg\varphi;$$
$$\exists\xi\,\varphi \leftrightarrow \neg\forall\xi\,\neg\varphi.$$

Напомним, что $\alpha \leftrightarrow \beta$ мы считаем сокращением для $(\alpha \to \beta) \wedge$ $\wedge\,(\beta \to \alpha)$, так что нам надо вывести четыре формулы.

Начнём с формулы $\exists\xi\,\varphi \to \neg\forall\xi\,\neg\varphi$. Имея в виду правило Бернайса, достаточно вывести формулу $\varphi \to \neg\forall\xi\,\neg\varphi$. Тавтология $(B \to \neg A) \to$ $\to (A \to \neg B)$ позволяет вместо этого выводить формулу $\forall\xi\,\neg\varphi \to \neg\varphi$, которая является аксиомой.

В только что выведенной формуле $\exists\xi\,\varphi \to \neg\forall\xi\,\neg\varphi$ можно в качестве φ взять любую формулу, в том числе начинающуюся с отрицания. Подставив $\neg\varphi$ вместо φ, получим

$$\exists\xi\,\neg\varphi \to \neg\forall\xi\,\neg\neg\varphi,$$

где $\neg\neg\varphi$ доказуемо эквивалентна φ и потому может быть заменена на φ. После этого правило контрапозиции (если из A следует $\neg B$, то из B следует $\neg A$) даёт

$$\forall\xi\,\varphi \to \neg\exists\xi\,\neg\varphi.$$

Выведем третью формулу: $\neg\exists\xi\,\neg\varphi \to \forall\xi\,\varphi$. По правилу Бернайса достаточно вывести $\neg\exists\xi\,\neg\varphi \to \varphi$, что после контрапозиции превращается в аксиому $\neg\varphi \to \exists\xi\,\neg\varphi$.

Четвёртая формула получится, если заменить в третьей φ на $\neg\varphi$ и применить контрапозицию.

4.2. Выводимость из посылок

В исчислении высказываний важную роль играло понятие выводимости из посылок и связанная с ним лемма о дедукции (с. 83). Для исчисления предикатов ситуация немного меняется. Если разрешить использовать посылки наравне с аксиомами безо всяких ограничений, то утверждение, аналогичное лемме о дедукции, будет неверным. Например, из формулы $A(x)$ можно вывести формулу $\forall x\,A(x)$ (как мы видели на с. 118 при обсуждении правила обобщения). Но импликация $A(x) \to \forall x\,A(x)$ не является выводимой (поскольку не общезначима).

Поэтому мы ограничимся случаем, когда все посылки являются замкнутыми формулами. Пусть Γ — произвольное множество замкнутых формул рассматриваемой нами сигнатуры σ. (Такие множества называют *теориями* в сигнатуре σ.) Говорят, что формула A *выводима из* Γ, если её можно вывести, используя наравне с аксиомами формулы из Γ. Как и для исчисления высказываний, мы пишем $\Gamma \vdash A$. Выводимые из Γ формулы называют также *теоремами* теории Γ.

Лемма о дедукции для исчисления предикатов. Пусть Γ — множество замкнутых формул, а A — замкнутая формула. Тогда $\Gamma \vdash (A \to B)$ тогда и только тогда, когда $\Gamma \cup \{A\} \vdash B$.

Доказательство проходит по той же схеме, что и для исчисления высказываний (с. 83): к формулам $C_1, \ldots, C_n$, образующим вывод $C_n = B$ из $\Gamma \cup \{A\}$, мы приписываем посылку A и дополняем полученную последовательность

$$(A \to C_1), \ldots, (A \to C_n)$$

до вывода из Γ. Отличие от пропозиционального случая в том, что в выводе могут встречаться правила Бернайса. Например, от выводимости формулы

$$A \to (\psi \to \varphi)$$

надо перейти к выводимости формулы

$$A \to (\psi \to \forall \xi\, \varphi)$$

(в которой переменная ξ не является параметром формулы ψ). Это несложно сделать, если заметить, что в силу пропозициональных тавтологий можно перейти от $A \to (\psi \to \varphi)$ к $(A \wedge \psi) \to \varphi$, затем применить правило Бернайса (это законно, так как переменная ξ не является параметром формулы ψ, а формула A замкнута по предположению). Получится выводимая из Γ формула

$$(A \wedge \psi) \to \forall \xi\, \varphi,$$

и остаётся вернуть A из конъюнкции в посылку.

Сходным образом рассматривается и второе правило Бернайса. Если выводима формула $A \to (\varphi \to \psi)$, то в силу пропозициональных тавтологий выводима формула $\varphi \to (A \to \psi)$, к которой можно применить правило Бернайса и получить $\exists \xi\, \varphi \to (A \to \psi)$, после чего вернуть A назад с помощью пропозициональной тавтологии. Лемма о дедукции доказана.

Отметим теперь несколько полезных свойств выводимости из посылок.

- Если $\Gamma \vdash A$ и $\Gamma' \supset \Gamma$, то $\Gamma' \vdash A$. (Очевидно следует из определения.)
- Если $\Gamma \vdash A$, то существует конечное множество $\Gamma' \subset \Gamma$, для которого $\Gamma' \vdash A$. (Вывод конечен и потому может использовать лишь конечное число формул.)

- Если Γ конечно и равно $\{\gamma_1, \ldots, \gamma_n\}$, то $\Gamma \vdash A$ равносильно выводимости (без посылок) формулы

$$(\gamma_1 \wedge \ldots \wedge \gamma_n) \to A.$$

  В самом деле, если $\{\gamma_1, \ldots, \gamma_n\} \vdash A$, то многократное применение леммы о дедукции даёт

$$\vdash \gamma_1 \to (\gamma_2 \to (\ldots(\gamma_n \to A)\ldots)),$$

  и остаётся воспользоваться надлежащей пропозициональной тавтологией. (В обратную сторону рассуждение также проходит без труда.)
- Комбинируя три предыдущих замечания, приходим к такому эквивалентному определению выводимости из посылок: $\Gamma \vdash A$, если найдутся формулы $\gamma_1, \ldots, \gamma_n \in \Gamma$, для которых

$$\vdash \gamma_1 \to (\gamma_2 \to (\ldots(\gamma_n \to A)\ldots)).$$

  Это определение имеет смысл и для формул с параметрами, так что если уж определять выводимость из посылок с параметрами (чего обычно избегают), то именно так.

Понятие выводимости из посылок позволяет переформулировать теорему о корректности исчисления предикатов.

Говорят, что интерпретация M сигнатуры σ является *моделью* теории Γ, если все формулы из Γ истинны в M.

Теорема 38 (о корректности; переформулировка). *Все теоремы теории* Γ *истинны в любой модели* M *теории* Γ.

Доказательство. Если формула A является теоремой теории Γ (т. е. $\Gamma \vdash A$), найдутся формулы $\gamma_1, \ldots, \gamma_n \in \Gamma$, для которых

$$\vdash \gamma_1 \to (\gamma_2 \to (\ldots(\gamma_n \to A)\ldots)).$$

По теореме о корректности (в уже известной нам форме) эта формула будет истинна во всех интерпретациях, в частности в M. Поскольку $\gamma_1, \ldots, \gamma_n$ истинны в M, то и формула A истинна в M (на любой оценке). $\square$

В следующих задачах — и только в них — знак $\vdash$ понимается в описанном выше смысле (в посылках допускаются параметры).

Задача 145. *Пусть* Γ *— множество произвольных (не обязательно замкнутых) формул.* **(а)** *Пусть существует "вывод" некоторой формулы* φ*, в котором наравне с аксиомами используются формулы из* Γ*, при этом все применения правил Бернайса предшествуют появлению формул из* Γ*. Покажите, что* $\Gamma \vdash \varphi$*. Покажите, что верно и обратное утверждение.*

(б) *Покажите, если в "выводе" формулы φ наравне с аксиомами используются формулы из Γ, но правила Бернайса не применяются по переменным, свободным в Γ, то $\Gamma \vdash \varphi$.*

Задача 146. *Покажите, что правила Бернайса можно переписать так:*

$$\frac{\Gamma, A \vdash B}{\Gamma, A \vdash \forall \xi\, B} \qquad \frac{\Gamma, B \vdash A}{\Gamma, \exists \xi\, \vdash A},$$

где переменная ξ не является параметром формулы A, а также параметром формул из Γ. (В первом правиле мы для симметрии выделили формулу A, хотя она ничем не отличается от формул из Γ.)

4.3. Переменные и константы

Отметим ещё несколько простых свойств выводимости, которые нам потребуются:

Лемма о свежих константах. Пусть выводима формула $\varphi(c/\xi)$, где φ — произвольная формула, ξ — переменная, c — константа, не входящая в формулу φ. Тогда выводима и формула φ.

Интуитивный смысл леммы: если мы доказали что-то про "свежую" константу c (не запятнавшую себя участием в формуле φ), то фактически мы доказали формулу φ для произвольных значений переменной.

Доказательство леммы. По условию существует вывод формулы $\varphi(c/\xi)$. Возьмём "свежую" переменную η, не встречающуюся в этом выводе, и всюду заменим в нём константу c на эту переменную. При этом вывод останется выводом, так как правила обращения с переменными и константами ничем не отличаются (кванторов по новой переменной в нём нет, так что корректные подстановки останутся корректными и применения правил Бернайса останутся допустимыми). Таким образом, выводима формула $\varphi(\eta/\xi)$.

По правилу обобщения выводима формула $\forall \eta\, \varphi(\eta/\xi)$. Осталось применить аксиому $\forall \eta\, \varphi(\eta/\xi) \to \varphi(\eta/\xi)(\xi/\eta)$; подстановка в правой части корректна и даёт формулу φ, так как сначала мы заменили свободные вхождения ξ на η, а затем обратно (так что в зону действия кванторов по ξ они попасть не могли). Лемма доказана.

Задача 147. *Сформулируйте и докажите аналогичную лемму для нескольких констант.*

Аналогичное рассуждение позволяет доказать и другое утверждение, которое нам потребуется:

Лемма о добавлении констант. Пусть формула φ некоторой сигнатуры σ выводима в исчислении предикатов расширенной сигнатуры σ', полученной из σ добавлением новых констант. Тогда φ выводима и в исчислении предикатов сигнатуры σ.

Доказательство. Пусть формула φ, не содержащая новых констант, имеет вывод, в котором новые константы встречаются. Как их оттуда удалить? Легко понять, что их можно заменить на свежие переменные, не входящие в вывод, и он останется выводом, но уже без новых констант. Лемма доказана.

На самом деле эта лемма верна для произвольного расширения сигнатуры (можно добавлять не только константы, но и функциональные символы любой валентности, а также предикатные символы). Чтобы удалить новые символы из вывода, поступаем так. Все термы вида $f(\ldots)$, где f — добавленный функциональный символ, мы заменяем на новую переменную (можно взять одну и ту же переменную для всех новых символов и всех их вхождений). Все атомарные формулы с новыми предикатными символами заменяем на какую-либо замкнутую формулу (одну и ту же; какая именно формула, роли не играет).

Задача 148. *Проведите это рассуждение подробно.*

Таким образом, мы можем говорить о выводимости формулы, не уточняя, в какой именно сигнатуре (содержащей все использованные в формуле предикатные и функциональные символы) мы ищем её вывод.

Если принять теорему о полноте, по которой выводимость равносильна общезначимости, независимость выводимости от сигнатуры становится очевидной: истинность формулы не зависит от интерпретации символов, которые в неё не входят. (Если интерпретировать отсутствующие в формуле символы как постоянные функции и предикаты, мы приходим к синтаксическому рассуждению, упомянутому выше.)

§5. Полнота исчисления предикатов

В этом разделе мы докажем, что всякая общезначимая формула выводима в исчислении предикатов. Мы будем следовать схеме, использованной в разделе 2, и введём понятия непротиворечивой и полной теории.

Фиксируем некоторую сигнатуру σ. Пусть Γ — теория в сигнатуре σ, то есть произвольное множество замкнутых формул этой сигнатуры. Говорят, что теория Γ *противоречива*, если в ней выводится некоторая формула φ и её отрицание $\neg\varphi$. В этом случае из Γ выводится любая формула, так как имеется аксиома $\neg A \to (A \to B)$. Если теория Γ не является противоречивой, то она называется *непротиворечивой*.

Задача 149. *Докажите, что теория противоречива тогда и только тогда, когда в ней выводится формула $\varphi \wedge \neg\varphi$ (здесь φ — произвольная формула сигнатуры).*

Непосредственно из определения следует, что всякое подмножество непротиворечивого множества непротиворечиво. Кроме того, если бесконечное множество противоречиво, то некоторое его конечное подмножество тоже противоречиво (поскольку в выводе участвует лишь конечное число формул).

Синтаксическое понятие непротиворечивости мы будем сравнивать с семантическим понятием совместности. Пусть имеется некоторая интерпретация M сигнатуры σ. Напомним, что она называется *моделью* теории Γ, если все формулы из Γ истинны в M. Множество Γ называется *совместным*, если оно имеет модель, то есть если все его формулы истинны в некоторой интерпретации.

Теорема 39 (о корректности; переформулировка). *Любое совместное множество замкнутых формул непротиворечиво.*

Доказательство. Пусть все формулы из Γ истинны в некоторой интерпретации M. Может ли оказаться, что $\Gamma \vdash \varphi$ и $\Gamma \vdash \neg\varphi$ для некоторой замкнутой формулы φ? Легко понять, что нет. В самом деле, в этом случае теорема 38 (с. 126) показывает, что формулы φ и $\neg\varphi$ должны быть одновременно истинны в M, что, очевидно, невозможно. □

Для доказательства обратного утверждения (о совместности непротиворечивой теории) нам понадобится понятие полной теории.

Непротиворечивое множество Γ, состоящее из замкнутых формул сигнатуры σ, называется *полным* в этой сигнатуре, если для любой замкнутой формулы φ этой сигнатуры либо формула φ, либо её отрицание $\neg\varphi$ выводятся из Γ.

Другими словами, теория полна, если из любых двух формул φ и $\neg\varphi$ (соответствующей сигнатуры) ровно одна является теоремой этой теории.

Полное множество можно получить, взяв какую-либо интерпретацию и рассмотрев все замкутые формулы, истинные в этой интерпретации. (Впоследствии мы увидим, что любое полное множество может быть получено таким способом — это легко следует из теоремы 40.)

В определении полноты существенно, что мы ограничиваемся замкнутыми формулами той же сигнатуры. Например, если мы возьмём одноместный предикатный символ S, не входящий в Γ, то формулы из Γ про него ничего не утверждают, и потому, скажем, ни формула $\forall x\, S(x)$, ни её отрицание не

выводимы из Γ. Замкнутость формулы φ тоже важна. Например, множество всех истинных в натуральном ряду формул сигнатуры $(=, <)$ полно, но ни формула $x = y$, ни формула $x \neq y$ из него не выводятся, иначе по правилу обобщения мы получили бы ложную в $\mathbb{N}$ формулу $\forall x \forall y\,(x = y)$ или $\forall x \forall y\,(x \neq y)$.

Полное множество подобно мировоззрению человека, достигшего предела умственного развития: на всё, что входит в круг его понятий (выражается формулой сигнатуры σ), он имеет точку зрения. Но это не относится ни к формулам большей сигнатуры (содержащим новые для него понятия), ни к формулам с параметрами (поскольку значения параметров не фиксированы).

Теперь мы готовы к доказательству основного результата этого раздела.

Теорема 40 (полнота исчисления предикатов, сильная форма). *Любая непротиворечивая теория совместна.*

Доказательство. Напомним, как мы доказывали аналогичное утверждение для высказываний. Мы расширяли наше непротиворечивое множество Γ до полного множества Γ′, а потом полагали пропозициональную переменную p истинной, если $\Gamma' \vdash p$. Здесь этого будет недостаточно, как мы увидим (например, непонятно, откуда брать носитель искомой модели). Но начало рассуждения будет таким же.

Лемма 1. Для всякого непротиворечивого множества Γ замкнутых формул сигнатуры σ существует полное непротиворечивое множество Γ′ замкнутых формул той же сигнатуры, содержащее Γ.

Доказательство повторяет рассуждение раздела 2: рассматривая по очереди замкнутые формулы, мы добавляем либо их, либо их отрицания в множество Γ.

Это можно сделать без труда для конечной или счётной сигнатуры (тогда множество всех замкнутых формул этой сигнатуры счётно). Лемма 1 доказана.

Как же нам теперь построить модель полного множества Γ? Прежде всего надо решить, что будет носителем этой модели. Заметим, что в сигнатуре могут быть некоторые константы (функциональные символы валентности 0). Им должны соответствовать некоторые элементы носителя. Кроме того, замкнутым термам (которые не содержат никаких переменных, только константы) также должны соответствовать элементы носителя. Попробуем взять в качестве носителя как раз множество T всех замкнутых термов нашей сигнатуры. При этом понятно, как надо определять сигнатурные функции на этом множестве: функция, соответствующая символу f валентности k, отображает замкнутые термы $t_1, \ldots, t_k$ в терм $f(t_1, \ldots, t_k)$. (Это

определение никак не зависит от Γ.)

Предикаты на этом множестве определяем так: если A — предикатный символ, а $t_1, \ldots, t_n$ — замкнутые термы, то предикат, соответствующий символу A, истинен на термах $t_1, \ldots, t_n$, если формула $A(t_1, \ldots, t_n)$ выводима из Γ.

Тем самым интерпретация полностью описана, и мы хотели бы доказать, что все формулы из Γ в ней истинны. Мы будем доказывать по индукции такой факт: если $\Gamma \vdash \varphi$, то формула φ истинна в построенной интерпретации, а если $\Gamma \vdash \neg\varphi$, то формула φ ложна.

Однако без дополнительных предположений о множестве Γ этот план обречён на неудачу, поскольку замкнутых термов может быть совсем мало (или даже вовсе не быть), в то время как соответствующая теория не имеет конечных моделей. Если начать индуктивное рассуждение, то выяснится, что трудность возникает в случае, когда формула φ начинается с квантора. Например, может оказаться, что формула $\exists x\, A(x)$ выводима из множества Γ, в то время как ни для какого замкнутого терма t формула $A(t)$ не выводима из Γ. Тогда формула $\exists x\, A(x)$ будет ложной в описанной нами модели (хотя выводимой). Чтобы преодолеть эту трудность, мы наложим дополнительные требования на множество Γ.

Назовём теорию (множество замкнутых формул сигнатуры σ) *экзистенциально полной* в сигнатуре σ, если для всякой формулы $\exists\xi\, \varphi$ сигнатуры σ, выводимой из Γ, найдётся замкнутый терм t этой сигнатуры, для которого $\Gamma \vdash \varphi(t/\xi)$.

Если множество Γ полно и экзистенциально полно, то описанная выше конструкция с замкнутыми термами даёт его модель. Прежде чем проверять это, покажем, как расширить Γ до полного и экзистенциально полного множества. Ключевую роль здесь играет такая лемма:

Лемма 2. Пусть Γ — непротиворечивое множество замкнутых формул, из которого выводится замкнутая формула $\exists\xi\, \varphi$. Пусть c — константа, не встречающаяся ни в Γ, ни в φ. Тогда множество Γ останется непротиворечивым после добавления формулы $\varphi(c/\xi)$.

(Замечание. Здесь и далее, говоря о непротиворечивости и выводимости, мы не уточняем, в какой сигнатуре строятся выводы: все наши сигнатуры будут отличаться лишь набором констант, и лемма о добавлении констант на с. 128 даёт нам такое право.)

Доказательство леммы 2. Пусть Γ становится противоречивым после добавления формулы $\varphi(c/\xi)$. Отсюда следует (используем подходящую пропозициональную тавтологию), что отрицание этой формулы выводится из Γ, то есть выводима формула $\gamma \to \neg\varphi(c/\xi)$, где γ — конъюнкция конечного числа формул из Γ. По лемме о свежих константах (с. 127) выводима фор-

мула $\gamma \to \neg\varphi$ (напомним, что c не входит ни в φ, ни в γ). Контрапозиция даёт формулу $\varphi \to \neg\gamma$, а правило Бернайса — формулу $\exists\xi\,\varphi \to \neg\gamma$. По предположению формула $\exists\xi\,\varphi$ выводима из Г, и множество Г оказывается противоречивым. Лемма 2 доказана.

Задача 150. *Докажите такое усиление леммы 2: при добавлении в* Г *формулы* $\varphi(c/\xi)$ *(в предположениях леммы) множество выводимых из* Г *формул исходной сигнатуры (без константы* c*) не меняется.*

Лемма 3. Пусть Г — непротиворечивое множество замкнутых формул сигнатуры σ. Тогда существует расширение сигнатуры σ новыми константами и непротиворечивое, полное и экзистенциально полное (в расширенной сигнатуре) множество Γ' замкнутых формул, содержащее Г.

Доказательство. Пусть сигнатура конечна или счётна. Тогда формул вида $\exists\xi\,\varphi$, выводимых из Г, не более чем счётное число. К каждой из них по очереди будем применять лемму 2, вводя новую константу. Согласно этой лемме, на каждом шаге множество Г остаётся непротиворечивым, поэтому оно будет непротиворечивым и после добавления счётного числа формул (вывод противоречия затрагивает лишь конечное число формул).

Однако нельзя утверждать, что полученное множество будет экзистенциально полным в новой сигнатуре, поскольку про формулы вида $\exists\xi\,\varphi$ с добавленными константами мы ничего не знаем. Пополним это множество, применив лемму 1, и повторим рассуждение: для каждой выводимой формулы, начинающейся с квантора существования, введём новую константу и т. д.

Затем снова пополним его, снова добавим константы, снова пополним и так сделаем счётное число раз. Объединение всех полученных множеств будет непротиворечивым, полным и экзистенциально полным. В самом деле, оно непротиворечиво, так как противоречие должно выводиться из конечного числа формул (и поэтому должно появиться уже на конечном шаге). Оно полно: любая замкнутая формула φ содержит конечное число новых констант, поэтому на каком-то шаге пополнения она или её отрицание станут выводимыми. Наконец, построенное множество экзистенциально полно по той же причине: всякая формула содержит конечное число новых констант, потому на следующем шаге для неё предусмотрена своя константа.

Лемма 3 доказана.

Последним шагом в доказательстве теоремы о полноте (всякое непротиворечивое множество замкнутых формул совместно) является такая лемма:

Лемма 4. Пусть Г — полное и экзистенциально полное множество замкнутых формул некоторой сигнатуры σ. Тогда существует интерпретация M сигнатуры σ, в которой истинны все формулы из Г.

Мы уже говорили, как надо строить такую интерпретацию. Повторим это более подробно. Рассмотрим все замкнутые термы сигнатуры σ, то есть термы, не содержащие переменных, а только функциональные символы и константы. (Такие термы существуют, поскольку теория Γ экзистенциально полна.) Это множество будет носителем интерпретации.

Как интерпретировать функциональные символы, понятно (это не зависит от множества Γ): если символ f имеет валентность n, то ему соответствует функция, которая отображает n замкнутых термов $t_1, \ldots, t_n$ в замкнутый терм $f(t_1, \ldots, t_n)$. Константы (функциональные символы валентности 0) интерпретируются сами собой.

Интерпретация предикатных символов такова. Пусть A — предикатный символ валентности n. Чтобы узнать, истинен ли соответствующий ему предикат на замкнутых термах $t_1, \ldots, t_n$, надо составить атомарную формулу $A(t_1, \ldots, t_n)$ и выяснить, что выводится из Γ — сама эта формула или её отрицание. (Здесь мы используем полноту.) В первом случае предикат будет истинным, во втором — ложным.

Индукцией по числу логических связок и кванторов в замкнутой формуле φ сигнатуры σ докажем такое утверждение:

$$\Gamma \vdash \varphi \Leftrightarrow \varphi \text{ истинна в } M.$$

Для атомарных формул это верно по построению интерпретации M.

Для пропозициональных связок рассуждение ничем не отличается от приведённого в разделе 2. Нам нужно проверить, что выводимость из Γ подчиняется тем же правилам, что и истинность:

$$\Gamma \vdash \neg A \Leftrightarrow \Gamma \nvdash A,$$
$$\Gamma \vdash A \wedge B \Leftrightarrow \Gamma \vdash A \text{ и } \Gamma \vdash B,$$
$$\Gamma \vdash A \vee B \Leftrightarrow \Gamma \vdash A \text{ или } \Gamma \vdash B,$$
$$\Gamma \vdash A \to B \Leftrightarrow \Gamma \nvdash A \text{ или } \Gamma \vdash B.$$

Все эти свойства несложно доказать. Первое из них выражает полноту (и непротиворечивость — напомним, что по определению полная теория всегда непротиворечива) множества Γ. Остальные свойства легко проверить, если иметь в виду, что все частные случаи пропозициональных тавтологий выводимы.

Пусть теперь формула φ имеет вид $\exists \xi\, \psi$, где ψ — формула с единственным параметром ξ (или без параметров). Предположим, что она выводима из Γ. Тогда в силу экзистенциальной полноты найдётся константа c, для которой $\Gamma \vdash \psi(c/\xi)$. Формула $\psi(c/\xi)$ имеет меньшее число логических связок, поэтому к ней можно применить предположение индукции и заключить, что

она истинна в M. Тогда формула ψ истинна на оценке $\xi \mapsto c$ (см. лемму 2 на с. 120 и замечание после неё), поэтому формула $\exists\xi\,\psi$ истинна в M.

Напротив, пусть формула $\exists\xi\,\psi$ истинна в M. Тогда (по определению истинности) найдётся элемент (замкнутый терм) t, для которого ψ истинна на оценке $\xi \mapsto t$ и потому формула $\psi(t/\xi)$ истинна в M. По предположению индукции формула $\psi(t/\xi)$ выводима из Γ. Осталось воспользоваться тем, что формула $\psi(t/\xi) \to \exists\xi\,\psi$ является аксиомой (напомним, подстановка замкнутого терма всегда корректна).

Наконец, рассмотрим случай, когда формула φ имеет вид $\forall\xi\,\psi$. Пусть она выводима из Γ. Формула $\forall\xi\,\psi \to \psi(t/\xi)$ является аксиомой для любого замкнутого терма t. Поэтому и формула $\psi(t/\xi)$ выводима из Γ. В ней меньше логических связок, чем в φ, поэтому по предположению индукции она истинна в M. Значит, формула ψ истинна на любой оценке $\xi \mapsto t$, и потому формула $\forall\xi\,\psi$ истинна в M.

Если формула $\forall\xi\,\psi$ не выводима из Γ, то из Γ выводится её отрицание. Оно, как мы видели, доказуемо эквивалентно формуле $\exists\xi\,\neg\psi$. Поэтому в силу экзистенциальной полноты выводима формула $\neg\psi(c/\xi)$ для некоторой константы c. Эта формула истинна, поэтому ψ ложна при некотором значении переменной ξ, так что формула $\forall\xi\,\psi$ ложна в M.

Таким образом, мы доказали, что всякое непротиворечивое множество замкнутых формул имеет модель (расширив его до полного и экзистенциально полного множества, у которого есть модель из замкнутых термов). □

Анализ доказательства позволяет сделать такое наблюдение:

Теорема 41. *Непротиворечивое множество замкнутых формул конечной или счётной сигнатуры имеет счётную модель.*

Доказательство. В самом деле, элементами построенной нами модели являются замкнутые термы, образованные из добавленных констант и функциональных символов сигнатуры. На каждом шаге добавляется счётное множество констант, поэтому всех констант счётное число, значит, и термов счётное число. □

Аналогичное рассуждение с использованием свойств операций с мощностями (о которых можно прочесть в [**1**]) устанавливает такой факт:

Теорема 42. *Всякое непротиворечивое множество формул сигнатуры σ имеет модель мощности* $\max(\aleph_0, |\sigma|)$ *(где $\aleph_0$ обозначает счётную мощность, а $|\sigma|$ — мощность сигнатуры).*

Кстати, при доказательстве теорем 41 и 42 можно было бы сослаться на теорему Левенгейма – Сколема об элементарной подмодели (построить модель произвольной мощности, а потом уменьшить, если надо).

Возвратимся теперь к исходной формулировке теоремы о полноте.

Теорема 43 (полнота исчисления предикатов, слабая форма). *Всякая общезначимая формула выводима в исчислении предикатов.*

Доказательство. Пусть формула φ замкнута. Если она невыводима, то множество $\{\neg\varphi\}$ непротиворечиво и потому совместно. В его модели формула φ будет ложной, что противоречит предположению.

Что касается незамкнутых формул, то их общезначимость и выводимость равносильна общезначимости и выводимости их замыкания. □

Как и в разделе 2, из теоремы о полноте можно вывести такое следствие:

Теорема 44 (компактность для исчисления предикатов). *Пусть* Γ *— множество замкнутых формул некоторой сигнатуры, и любое его конечное подмножество имеет модель. Тогда и само множество* Γ *имеет модель.*

Доказательство. В самом деле, по теореме о полноте (и корректности, если быть точным) наличие модели (совместность) равносильно непротиворечивости. А по определению противоречивость затрагивает лишь конечное число формул из Γ. □

Задача 151. *Покажите, что теорема о полноте в сильной форме является следствием теоремы компактности и теоремы о полноте в слабой форме. (Указание: если множество* Γ *не имеет модели, то его конечная часть не имеет модели, поэтому формула* $\langle\ldots\rangle$ *общезначима, поэтому . . .)*

Ещё один важный результат, вытекающий из теоремы о полноте — совпадение синтаксического понятия выводимости и семантического понятия следования. Пусть дана некоторая сигнатура σ. Рассмотрим множество Γ замкнутых формул этой сигнатуры (такие множества мы называем теориями в сигнатуре σ) и ещё одну замкнутую формулу φ. Говорят, что φ *семантически следует* из Γ, если φ истинна во всякой модели теории Γ, то есть во всякой интерпретации сигнатуры σ, где истинны все формулы из Γ. (Обозначение: $\Gamma \models \varphi$.)

Теорема 45.

$$\Gamma \vdash \varphi \Leftrightarrow \Gamma \models \varphi.$$

Доказательство. Если $\Gamma \vdash \varphi$, то $\Gamma \models \varphi$ (как мы видели в теореме 38 на с. 126).

Напротив, пусть φ не выводима из Γ. Тогда теория $\Gamma \cup \{\neg\varphi\}$ непротиворечива и (в силу теоремы о полноте) имеет модель. Значит φ не следует из Γ. □

Задача 152. *Какими нужно взять φ и Γ в этой теореме, чтобы получить приведённые ранее формулировки теоремы о полноте? (Ответ: при $\varphi = \bot$ (тождественно ложная формула) получаем сильную форму теоремы о полноте, при $\Gamma = \varnothing$ — слабую.)*

§6. О выводах и доказательствах

Центральная идея математической логики восходит ещё к Лейбницу и состоит в том, чтобы записывать математические утверждения в виде последовательностей символов и оперировать с ними по формальным правилам. При этом правильность рассуждений можно проверять механически, не вникая в их смысл.

Усилиями большого числа математиков и логиков второй половины XIX и первой половины XX века (Буль, Кантор, Фреге, Пеано, Рассел, Уайтхед, Цермело, Френкель, Гильберт, фон Нейман, Гёдель и другие) эта программа была в основном выполнена. Принято считать, что всякое точно сформулированное математическое утверждение можно записать формулой теории множеств (одной из наиболее общих формальных теорий), а всякое строгое математическое доказательство преобразовать в формальный вывод в этой теории (последовательность формул теории множеств, подчиняющуюся некоторым простым правилам). В каком-то смысле это даже стало определением: математически строгим считается такое рассуждение, которое можно перевести на язык теории множеств.

Так что же, теперь математики могут дружно уйти на пенсию, поскольку можно открывать математические теоремы с помощью компьютеров, запрограммированных в соответствии с формальными правилами теории множеств? Конечно, нет, причём сразу по нескольким причинам.

Начнём с того, что машина, выдающая с большой скоростью математические теоремы (и их доказательства), хотя и возможна, но бесполезна. Дело в том, что среди этих верных утверждений почти все будут неинтересными. Формальная логика говорит, какие правила надо соблюдать, чтобы получать верные результаты, но не говорит, в каком порядке их надо применять, чтобы получить что-то интересное.

Казалось бы, мы можем запустить машину и ждать, пока она не докажет интересующее нас утверждение (пропуская все остальные). Проблема в том, что формальное доказательство сколько-нибудь содержательной теоремы настолько длинно, что прочесть его человек не в состоянии. Представьте себе доказательство, которое состоит из миллионов формально правильных шагов, в котором мы можем проверить каждый отдельный шаг, но так и

не понимаем, что происходит — много ли в нём проку?

На самом деле прок всё-таки есть: мы узнаём, что доказываемое утверждение верно, хотя так и не понимаем, почему. Так что и такая машина была бы полезна. Увы, и этого сделать не удаётся, поскольку на поиск доказательства сколько-нибудь сложного утверждения известными сейчас методами требуется астрономически большое время (даже если представить себе, что машина работает с предельно возможной по законам физики скоростью).

Можно умерить амбиции и поставить более простую задачу: пусть машина проверяет доказательства, записанные человеком по правилам формальной логики. Если машина не может помочь нам что-то открыть, пусть она хотя бы проверит, не пропустили ли мы какого-то шага рассуждения.

Из всех перечисленных задач эта выглядит наиболее реалистичной. К сожалению, пока что работы и в этом направлении не ушли далеко: формальная запись доказательства в виде, пригодном для машинной проверки, является долгим и скучным делом, на которое у математиков не хватает энтузиазма и терпения. А разработать удобные средства такой записи пока не удалось.

Короче говоря, революционная программа Лейбница построения формальных оснований математики осуществилась, но незаметно: под здание математики подвели новый (и довольно прочный) фундамент, но большинство жильцов про это до сих пор не знают.

Так что же, математическая логика бесполезна? Ни в коем случае: она не только удовлетворяет естественный философский интерес к основаниям математики, но и содержит множество красивых результатов, которые важны не только для математики, но и для computer science.

В качестве иллюстрации к этим общим соображениям рассмотрим так называемую проблему четырех красок.

Раскрашивая географическую карту естественно пользоваться по возможности меньшим количеством цветов, однако так, чтобы две страны, имеющих общую часть границы (не только общую точку), были бы окрашены по разному. В 1852 году Френсис Гутри, составляя карту графств Англии обратил внимание, что для такой цели вполне хватает четырех красок. Его брат, Фредерик, сообщил об этом наблюдении известному математику ДеМоргану, который сделал эту гипотезу достоянием математической общественности. Первая точная формулировка в печати принадлежит Кэли (1878).

Первое (неправильное) доказательство появилось год спустя и принадлежало Кемпе. Ошибку в нем обнаружил Хивуд целых одиннадцать лет спустя. Однако, просматривая рассуждения Кели, Хивуд понял, что из них

следует, что пяти красок действительно хватает. Тем не менее, для любой конкретной карты хватало-таки четырех красок! За первым неправильным доказательство последовало множество других. В этом отношении проблема четырех красок уступала лишь знаменитой проблеме Ферма.

До середины этого века, несмотря на то, что проблемой четырех красок занимались многие выдающиеся математики, положение с доказательством изменилось несущественно. Наиболее значительный вклад в решение проблемы внес в 1913 году Биркгоф, чьи идеи позволили Франклину доказать гипотезу для карты с не более чем двадцатью пятью странами. Позже было установлено, что если число стран не превосходит тридцати восьми, то гипотеза справедлива. Из чего было понятно что доказательство (или опровергающий пример) не могут быть особенно просты.

В 1977 году доказательство гипотезы четырех красок было наконец получено Апелем и Хакеном и опубликовано в двух статьях в журнале Contemporary Mathematics. Весьма значительная часть рутинных проверок была выполнена компьютером, и это революционное нововведение в сложившуюся практику дедуктивных рассуждений в чистой математике служит основанием для некоторого естественного скептицизма по отношению к данному доказательству и по сей день.

Между тем, для человека поверхностно знакомого с математикой, предыдущая фраза должна вызвать изумление: а как же обязательный для математики, и только для нее, принцип *tertium non datur* (исключенного третьего) в соответствии с которым утверждение либо справедливо, либо нет? Дело обстоит не так просто. Вот что пишут сами авторы доказательства:

"Читатель должен разобраться в 50 страницах текста и диаграмм, 85 страницах с почти 2500 дополнительными диаграммами, с 400 страницами микрофишей содержащими еще диаграммы, а также тысячи отдельных проверок утверждений, сделанных в 24 леммах основного текста. Вдобавок читатель узнает, что проверка некоторых фактов потребовала 1200 часов компьютерного времени и которые были бы чрезвычайно длительными при проверке вручную. Статьи устрашающи по стилю и и длине, и немногие математики прочли их сколько-нибудь подробно."

Говоря прямо, компьютерную часть доказательства невозможно проверить вручную, а традиционная часть доказательства длинна и сложна настолько, что ее никто целиком и не проверял. Между тем, доказательство, не поддающееся проверке, есть нонсенс. Согласиться с подобным доказательством означает примерно то же, что просто поверить авторам. Но и здесь не все так просто.

Многие элементарные рассуждения — например, формула Эйлера о гра-

фах, входящая в доказательство, основана на утверждениях типа: "Плоский граф разрезает плоскость на совокупность $D(G)$ неперекрывающихся многоугольных областей", что вроде бы нетрудно проверить, взяв лист бумаги и карандаш.. К сожалению, это утверждение не принадлежит к числу аксиом или школьных теорем плоской геометрии и его нужно доказывать. Соответствующая теорема, сформулированная Жорданом, доказывается очень непросто, однако основные трудности связаны с тем, как следует понимать слова типа "разрезает", интуитивно вполне ясные, но с трудом поддающимся формализации. В свете этого замечания становится уже не совсем понятным, доказана ли формула Эйлера или мы *поверили* правдоподобным рассуждениям, основанным на интуитивных представлениях о свойствах геометрических фигур.

Долгое время идеалом математической строгости были формулировки и доказательства Эвклида, в которых осуществлялась программа вывода теорем из аксиом по определенным правилам (так называемый метод дедукции, позволяющий получать неочевидные утверждения из очевидных посредством ряда явно предъявляемых элементарных, очевидно законных, умозаключений). Этот образец строгости и в наше время недостижим в курсе школьной математики, но для современной чистой математики стандарты Эвклида недостаточны. Эвклид по-видимому не задумывался о том, почему прямая делит плоскость на две части (и что это значит), он не определял понятия "между", считая это понятие очевидным и т.д. (Большая часть соответствующих утверждений включена в аксиоматику геометрии только в последнюю сотню лет). Формальные выводы из новой системы аксиом стали гораздо длиннее, чем в античные времена. Трудно даже вообразить длину полного вывода формулы Эйлера в соответствии с современными стандартами математической логики и системы аксиом геометрии.

Но что совершенно точно, так это то, что это такое рассуждение никогда не было проделано из-за бесполезности этого занятия: формальные выводы практически не поддаются проверке в силу свойств человеческой психики: помимо их гораздо большей (по сравнению с привычными рассуждениями) длины их сознательное усвоение идет гораздо медленнее. Поэтому обычно удовлетворяются *уверенностью* в том, что формальный вывод возможен в принципе.

В формуле Эйлера, например, математики не сомневаются. Вообще, принятие доказательства есть некий социальный акт. Выдающийся алгебраист Ю.И. Манин в своей книге "Доказуемое и недоказуемое" [**5**] пишет по этому поводу: "... отсутствие ошибок в математической работе (если они не обнаружены), как и в других естественных науках, часто устанавливается по косвенным данным: имеет значение соответствие с общими ожиданиями,

использование аналогичных аргументов в других работах, разглядывание "под микроскопом" отдельных участков доказательства, даже репутация автора; словом, воспроизводимость в широком смысле. "Непонятные" доказательства могут сыграть очень полезную роль, стимулируя поиски более доступных рассуждений."

Именно такая история происходит и с доказательством теоремы о четырех красках. Не так давно появилось новое доказательство, причем та часть, которая выполнена не на компьютере, уже поддается проверке. Однако компьютерная часть все еще остается скорее предметом веры. Ведь даже проверка распечаток всех программ и всех входных данных не может гарантировать от случайных сбоев или даже от скрытых пороков электроники (вспомним, что неумение делить первой версии процессора Pentium было случайно обнаружено спустя полгода после начала его коммерческих продаж — кстати, "чистым" математиком, работавшим в области теории чисел). По-видимому, единственный способ проверки компьютерных результатов — написать другую программу и для другого типа компьютера. Это, конечно, совсем непохоже на стандартный идеал дедуктивных рассуждений, но именно так осуществляется проверка утверждений во всех экспериментальных науках.

Из которых математика, стало быть, исключена напрасно.

ГЛАВА VII

Вычислимые функции, разрешимые и перечислимые множества

§1. Вычислимые функции

Функция f с натуральными аргументами и значениями называется *вычислимой*, если существует алгоритм, её вычисляющий, то есть такой алгоритм A, что

- если $f(n)$ определено для некоторого натурального n, то алгоритм A останавливается на входе n и печатает $f(n)$;
- если $f(n)$ не определено, то алгоритм A не останавливается на входе n.

Несколько замечаний по поводу этого определения:

1. Понятие вычислимости определяется здесь для частичных функций (областью определения которых является некоторое подмножество натурального ряда). Например, нигде не определённая функция вычислима (в качестве A надо взять программу, которая всегда зацикливается).

2. Можно было бы изменить определение, сказав так: "если $f(n)$ не определено, то либо алгоритм A не останавливается, либо останавливается, но ничего не печатает". На самом деле от этого ничего бы не изменилось (вместо того, чтобы останавливаться, ничего не напечатав, алгоритм может зацикливаться).

3. Входами и выходами алгоритмов могут быть не только натуральные числа, но и двоичные строки (слова в алфавите $\{0,1\}$), пары натуральных чисел, конечные последовательности слов и вообще любые, как говорят, "конструктивные объекты". Поэтому аналогичным образом можно определить понятие, скажем, вычислимой функции с двумя натуральными аргументами, значениями которой являются рациональные числа.

Для функций, скажем, с действительными аргументами и значениями понятие вычислимости требует специального определения. Здесь ситуация сложнее, определения могут быть разными, и мы о вычислимости таких функций говорить не будем. Отметим только, что, например, синус (при разумном определении вычислимости) вычислим, а функция $\mathrm{sign}(x)$, равная -1, 0 и 1 при $x < 0$, $x = 0$ и $x > 0$ соответственно — нет. Точно так же требует специального определения вычислимость функций, аргументами

которых являются бесконечные последовательности нулей и единиц и т. п.

4. Несколько десятилетий назад понятие алгоритма требовало специального разъяснения. Сейчас ("компьютерная грамотность"?) такие объяснения всё равно никто читать не будет, поскольку и так ясно, что такое алгоритм. Но всё же надо соблюдать осторожность, чтобы не принять за алгоритм то, что им не является. Вот пример неверного рассуждения:

"Докажем", что всякая вычислимая функция f с натуральными аргументами и значениями может быть продолжена до всюду определённой вычислимой функции $g\colon \mathbb{N} \to \mathbb{N}$. В самом деле, если f вычисляется алгоритмом A, то следующий алгоритм B вычисляет функцию g, продолжающую f: "если A останавливается на n, то B даёт тот же результат, что и A; если A не останавливается на n, то B даёт результат (скажем) 0". (В чём ошибка в этом рассуждении?)

§2. Разрешимые множества

Множество натуральных чисел X называется *разрешимым*, если существует алгоритм, который по любому натуральному n определяет, принадлежит ли оно множеству X.

Другими словами, X разрешимо, если его *характеристическая функция* $\chi(n) = (\textbf{if } n \in X \textbf{ then } 1 \textbf{ else } 0 \textbf{ fi})$ вычислима.

Очевидно, пересечение, объединение и разность разрешимых множеств разрешимы. Любое конечное множество разрешимо.

Аналогично определяют разрешимость множеств пар натуральных чисел, множеств рациональных чисел и т. п.

Задача 153. *Докажите, что множество всех рациональных чисел, меньших числа e (основания натуральных логарифмов), разрешимо.*

Задача 154. *Докажите, что непустое множество натуральных чисел разрешимо тогда и только тогда, когда оно есть множество значений всюду определённой неубывающей вычислимой функции с натуральными аргументами и значениями.*

Отметим тонкий момент: можно доказать разрешимость множества неконструктивно, не предъявляя алгоритма. Вот традиционный пример: множество тех n, для которых в числе π есть не менее n девяток подряд, разрешимо. В самом деле, это множество содержит либо все натуральные числа, либо все натуральные числа вплоть до некоторого. В обоих случаях оно разрешимо. Тем не менее мы так и не предъявили алгоритма, который по n узнавал бы, есть ли в π не менее n девяток подряд.

Задача 155. *Использованы ли в этом рассуждении какие-то свойства числа π? Что изменится, если заменить слова "не менее n девяток" на "ровно n девяток (окружённых не-девятками)"?*

Существуют ли неразрешимые множества? Существуют — просто потому, что алгоритмов (и поэтому разрешимых подмножеств натурального ряда) счётное число, а всех подмножеств натурального ряда несчётное число. Более конкретные примеры мы ещё построим.

§3. Перечислимые множества

Множество натуральных чисел называется *перечислимым*, если оно перечисляется некоторым алгоритмом, то есть если существует алгоритм, который печатает (в произвольном порядке и с произвольными промежутками времени) все элементы этого множества и только их.

Такой алгоритм не имеет входа; напечатав несколько чисел, он может надолго задуматься и следующее число напечатать после большого перерыва (а может вообще больше никогда ничего не напечатать — тогда множество будет конечным).

Существует много эквивалентных определений перечислимого множества. Вот некоторые из них:

1) Множество перечислимо, если оно есть область определения вычислимой функции.
2) Множество перечислимо, если оно есть область значений вычислимой функции.
3) Множество X перечислимо, если его (как иногда говорят) "полухарактеристическая" функция, равная 0 на элементах X и не определённая вне X, вычислима.

Чтобы доказать эквивалентность этих определений, воспользуемся возможностью пошагового исполнения алгоритма.

Пусть X перечисляется некоторым алгоритмом A. Покажем, что полухарактеристическая функция множества X вычислима. В самом деле, алгоритм её вычисления таков: получив на вход число n, пошагово выполнять алгоритм A, ожидая, пока он напечатает число n. Как только он это сделает, выдать на выход 0 и закончить работу.

Наоборот, пусть X есть область определения (вычислимой) функции f, вычисляемой некоторым алгоритмом B. Тогда X перечисляется таким алгоритмом A: ßПараллельно запускать B на входах $0, 1, 2, \ldots$, делая всё больше шагов работы алгоритма B (сначала один шаг работы на входах 0 и 1; потом по два шага работы на входах $0, 1, 2$, потом по три на входах $0, 1, 2, 3$

и так далее). Все аргументы, на которых алгоритм B заканчивает работу, печатать по мере обнаружения.

Итак, мы установили эквивалентность исходного определения определениям 1 и 3. Если в только что приведённом описании алгоритма A печатать не аргументы, на которых B заканчивает работу, а результаты этой работы, то получается алгоритм, перечисляющий область значений функции f. Осталось ещё убедиться, что всякое перечислимое множество есть область значений вычислимой функции. Это можно сделать, например, так: пусть X есть область определения вычислимой функции, вычисляемой некоторым алгоритмом A. Тогда X есть область значений функции

$$b(x) = \begin{cases} x, \text{ если } A \text{ заканчивает работу на } x, \\ \text{не определено в противном случае.} \end{cases}$$

Вычисляющий эту функцию алгоритм действует так же, как и A, но только вместо результата работы алгоритма A выдаёт копию входа.

Ещё одно эквивалентное определение перечислимого множества: множество натуральных чисел перечислимо, если оно либо пусто, либо есть множество значений всюду определённой вычислимой функции (другими словами, его элементы можно расположить в вычислимую последовательность).

В самом деле, пусть перечислимое множество X, перечисляемое алгоритмом A, непусто. Возьмём в нём какой-то элемент x_0. Теперь рассмотрим такую всюду определённую функцию a: если на n-м шаге работы алгоритма A появляется число t, то положим $a(n) = t$; если же ничего не появляется, то положим $a(n) = x_0$. (Мы предполагаем, что на данном шаге работы алгоритма может появиться только одно число — в противном случае работу надо разбить на более мелкие шаги.)

Заметим, что это рассуждение неконструктивно — имея алгоритм A, мы можем не знать, пусто ли перечисляемое им множество или нет.

Теорема 46. *Пересечение и объединение перечислимых множеств перечислимы.*

Доказательство. Если X и Y перечисляются алгоритмами A и B, то их объединение перечисляется алгоритмом, который параллельно выполняет по шагам A и B и печатает всё, что печатают A и B. С пересечением немного сложнее — результаты работы A и B надо накапливать и сверять друг с другом; что появится общего — печатать. □

Задача 156. *Проведите это рассуждение, используя какое-либо другое эквивалентное определение перечислимости.*

Как мы увидим, дополнение перечислимого множества не обязано быть перечислимым.

Задача 157. *Иногда говорят о так называемых "недетерминированных алгоритмах" (оксюморон, но распространённый) — такой алгоритм включает в себя команды типа*

$$n := \text{произвольное натуральное число}$$

(достаточно, впрочем, команды "$n := 0$ или 1", так как произвольное число можно формировать по битам). Недетерминированный алгоритм (при одном и том же входе) может действовать по-разному, в зависимости от того, какие "произвольные" числа будут выбраны. Докажите, что перечислимое множество можно эквивалентно определить как множество чисел, которые могут появиться на выходе недетерминированного алгоритма (при фиксированном входе).

Задача 158. *Докажите, что если множества $A \subset \mathbb{N}$ и $B \subset \mathbb{N}$ перечислимы, то их декартово произведение $A \times B \subset \mathbb{N} \times \mathbb{N}$ также перечислимо.*

§4. Перечислимые и разрешимые множества

Теорема 47. *Всякое разрешимое множество натуральных чисел перечислимо. Если множество A и его дополнение (до множества всех натуральных чисел) перечислимы, то A разрешимо.*

Доказательство. Если принадлежность числа к множеству A можно проверить некоторым алгоритмом, то A и его дополнение перечислимы: надо по очереди проверять принадлежность чисел $0, 1, 2, \ldots$ и печатать те из них, которые принадлежат A (или те, которые не принадлежат A).

В другую сторону: если у нас есть алгоритм, перечисляющий A, а также другой алгоритм, перечисляющий дополнение к A, то для выяснения принадлежности заданного числа n к A надо запустить оба эти алгоритма и ждать, пока один из них напечатает n (мы знаем, что рано или поздно ровно один из них это сделает). Посмотрев, какой алгоритм это сделал, мы узнаем, лежит ли n в A. □

Этот факт называют *теоремой Пóста*.

Она говорит, что разрешимые множества — это перечислимые множества с перечислимыми дополнениями. Напротив, перечислимые множества можно определить через разрешимые:

Теорема 48. *Множество P натуральных чисел перечислимо тогда и только тогда, когда оно является проекцией некоторого разрешимого*

множества Q пар натуральных чисел. (Проекция получается, если от пар оставить их первые компоненты: $x \in P \Leftrightarrow \exists y(\langle x, y\rangle \in Q)$.)

Доказательство. Проекция любого перечислимого множества перечислима (перечисляющий алгоритм должен лишь удалять вторые члены пар), так что проекция разрешимого множества тем более перечислима.

Напротив, если P — перечислимое множество, перечисляемое алгоритмом A, то оно есть проекция разрешимого множества Q, состоящего из всех таких пар $\langle x, n\rangle$, что x появляется в течении первых n шагов работы алгоритма A. (Это свойство, очевидно, разрешимо.) □

§5. Перечислимость и вычислимость

Мы видели, что перечислимое множество можно определить в терминах вычислимых функций (например, как область определения вычислимой функции). Можно сделать и наоборот:

Теорема 49. *Функция f с натуральными аргументами и значениями вычислима тогда и только тогда, когда её график*

$$F = \{\langle x, y\rangle \mid f(x) \text{ определено и равно } y\}$$

является перечислимым множеством пар натуральных чисел.

Доказательство. Пусть f вычислима. Тогда существует алгоритм, перечисляющий её область определения, то есть печатающий все x, на которых f определена. Если теперь для каждого из таких x вычислять ещё и значение $f(x)$, получим алгоритм, перечисляющий множество F.

Напротив, если имеется алгоритм, перечисляющий F, то функция f вычисляется таким алгоритмом: имея на входе n, ждём появления в F пары, первый член которой равен n; как только такая пара появилась, печатаем её второй член и кончаем работу. □

Пусть f — частичная функция с натуральными аргументами и значениями. *Образ* множества A при f определяется как множество всех чисел $f(n)$, для которых $n \in A$ и $f(n)$ определено. *Прообраз* множества A при f определяется как множество всех тех n, при которых $f(n)$ определено и принадлежит A.

Теорема 50. *Прообраз и образ перечислимого множества при вычислимой функции перечислимы.*

Доказательство. В самом деле, прообраз перечислимого множества A при вычислимой функции f можно получить так: взять график f, пересечь его с перечислимым множеством $\mathbb{N} \times A$ и спроектировать на первую координату. Рассуждение для образов аналогично, только координаты меняются местами. □

Задача 159. *Пусть F — перечислимое множество пар натуральных чисел. Докажите. что существует вычислимая функция f, определённая на тех и только тех x, для которых найдётся y, при котором $\langle x, y\rangle \in F$, причём значение $f(x)$ является одним из таких y. (Это утверждение называют иногда* теоремой об униформизации.

Задача 160. *Даны два пересекающихся перечислимых множества X и Y. Докажите, что найдутся непересекающиеся перечислимые множества $X' \subset X$ и $Y' \subset Y$, для которых $X' \cup Y' = X \cup Y$.*

Задача 161. Диофантовым *называется уравнение, которое имеет вид $P(x_1, \ldots, x_n) = 0$, где P — многочлен с целыми коэффициентами. Докажите, что множество диофантовых уравнений, имеющих целые решения, перечислимо. (Оно неразрешимо: в этом состоит известный результат Ю. В. Матиясевича, явившийся решением знаменитой "10-й проблемы Гильберта".)*

Задача 162. *Не ссылаясь на доказательство теоремы Ферма, покажите, что множество всех показателей n, для которых существует решение уравнения $x^n + y^n = z^n$ в целых положительных числах, перечислимо. (Как теперь известно, это множество содержит лишь числа 1 и 2.)*

Задача 163. *Покажите, что всякое бесконечное перечислимое множество можно записать в виде $\{a(0), a(1), a(2), \ldots\}$, где a — вычислимая функция, все значения которой различны. (Указание: в ходе перечисления удаляем повторения.)*

Задача 164. *Покажите, что всякое бесконечное перечислимое множество содержит бесконечное разрешимое подмножество. (Указание: воспользуемся предыдущей задачей и выберем возрастающую подпоследовательность.)*

Задача 165. *Покажите, что для всякой вычислимой функции f существует вычислимая функция, являющаяся "псевдообратной" к f в следующем смысле: область определения g совпадает с областью значений f, и при этом $f(g(f(x))) = f(x)$ для всех x, при которых $f(x)$ определено.*

Задача 166. *Действительное число α называется* вычислимым, *если существует вычислимая функция a, которая по любому рациональному $\varepsilon > 0$ даёт рациональное приближение к α с ошибкой не более ε, т. е. $|\alpha - a(\varepsilon)| \leqslant \varepsilon$ для любого рационального $\varepsilon > 0$. (Рациональное число является конструктивным объектом, так что понятие вычислимости не требует специального уточнения.)*

(**а**) *Докажите, что число α вычислимо тогда и только тогда, когда множество рациональных чисел, меньших α, разрешимо.*

(**б**) *Докажите, что число α вычислимо тогда и только тогда, когда последовательность знаков представляющей его десятичной (или двоичной) дроби вычислима.*

(**в**) *Докажите, что число α вычислимо тогда и только тогда, когда существует вычислимая последовательность рациональных чисел, вычислимо сходящаяся к α (последнее означает, что можно алгоритмически указать N по ε в стандартном ε-N-определении сходимости.)*

(**г**) *Покажите, что сумма, произведение, разность и частное вычислимых действительных чисел вычислимы. Покажите, что корень многочлена с вычислимыми коэффициентами вычислим.*

(**д**) *Сформулируйте и докажите утверждение о том, что предел вычислимо сходящейся последовательности вычислимых действительных чисел вычислим.*

(**е**) *Действительное число α называют* перечислимым снизу, *если множество всех рациональных чисел, меньших α, перечислимо. (Перечислимость сверху определяется аналогично.) Докажите, что число α перечислимо снизу тогда и только тогда, когда оно является пределом некоторой вычислимой возрастающей последовательности рациональных чисел.*

(**ж**) *Докажите, что действительное число вычислимо тогда и только тогда, когда оно перечислимо снизу и сверху.*

Дальнейшие свойства вычислимых действительных чисел см. в задаче 175.

ГЛАВА VIII

Универсальные функции и неразрешимость

§1. Универсальные функции

Сейчас мы построим пример перечислимого множества, не являющегося разрешимым. При этом будет использоваться так называемая универсальная функция.

Говорят, что функция U двух натуральных аргументов является *универсальной* для класса вычислимых функций одного аргумента, если для каждого n функция

$$U_n\colon x \mapsto U(n, x)$$

("сечение" функции U при фиксированном n) является вычислимой и если все вычислимые функции (одного аргумента) встречаются среди U_n. (Напомним, что ни функция U, ни вычислимые функции одного аргумента не обязаны быть всюду определёнными.)

Аналогичное определение можно дать и для других классов функций (одного аргумента): например, функция U двух аргументов будет универсальной для класса всех всюду определённых вычислимых функций одного аргумента, если её сечения U_n являются всюду определёнными вычислимыми функциями одного аргумента и исчерпывают все такие функции. Очевидно, универсальные функции существуют для любых счётных классов (и только для них).

Ключевую роль в этом разделе играет такой факт:

Теорема 51. *Существует вычислимая функция двух аргументов, являющаяся универсальной функцией для класса вычислимых функций одного аргумента.*

Доказательство. Запишем все программы, вычисляющие функции одного аргумента, в вычислимую последовательность $p_0, p_1, \ldots$ (например, в порядке возрастания их длины). Положим $U(i, x)$ равным результату работы i-ой программы на входе x. Тогда функция U и будет искомой вычислимой универсальной функцией. Сечение U_i будет вычислимой функцией, вычисляемой программой p_i. Алгоритм, вычисляющий саму функцию U, есть по существу интерпретатор для используемого языка программирования (он

применяет первый аргумент ко второму, если отождествить программу и её номер). □

Задача 167. *Все сечения U_n некоторой функции U двух аргументов вычислимы. Следует ли отсюда, что функция U вычислима?*

Задача 168. *Дайте (естественное) определение понятия вычислимой функции трёх аргументов, универсальной для класса вычислимых функций двух аргументов, и докажите её существование.*

Для множеств используется аналогичная терминология: множество $W \subset \subset \mathbb{N} \times \mathbb{N}$ называют *универсальным* для некоторого класса множеств натуральных чисел, если все сечения

$$W_n = \{x \mid \langle n, x\rangle \in W\}$$

множества W принадлежат этому классу и других множеств в классе нет.

Теорема 52. *Существует перечислимое множество пар натуральных чисел, универсальное для класса всех перечислимых множеств натуральных чисел.*

Доказательство. Рассмотрим область определения универсальной функции U. Она будет универсальным перечислимым множеством, поскольку всякое перечислимое множество является областью определения некоторой вычислимой функции U_n. □

Задача 169. *Как построить универсальное множество, исходя из того, что всякое перечислимое множество есть множество значений некоторой функции U_n?*

Задача 170. *Существует ли разрешимое множество пар натуральных чисел, универсальное для класса всех разрешимых множеств натуральных чисел?*

§2. Диагональная конструкция

В предыдущем разделе мы построили универсальную функцию для класса всех вычислимых функций одного аргумента. Можно ли сделать то же самое для класса всюду определённых вычислимых функций? Оказывается, что нет.

Теорема 53. *Не существует вычислимой всюду определённой функции двух аргументов, универсальной для класса всех вычислимых всюду определённых функций одного аргумента.*

Доказательство. Воспользуемся "диагональной конструкцией" — точно так же доказывается несчётность множества всех бесконечных десятичных дробей. Пусть U — произвольная вычислимая всюду определённая функция двух аргументов. Рассмотрим диагональную функцию $u(n) = U(n, n)$. Очевидно, на аргументе n функция u совпадает с функцией U_n, а функция $d(n) = u(n) + 1$ отличается от U_n. Таким образом, вычислимая всюду определённая функция $d(n)$ отличается от всех сечений U_n, и потому функция U не является универсальной. □

Почему это рассуждение не проходит для класса всех вычислимых функций (в том числе частичных)? Дело в том, что значение $d(n) = U(n, n) + 1$ теперь не обязано отличаться от значения $U_n(n) = U(n, n)$, так как оба они могут быть не определены.

Тем не менее, часть рассуждения остаётся в силе.

Теорема 54. *Существует вычислимая функция d (с натуральными аргументами и значениями), от которой никакая вычислимая функция f не может всюду отличаться: для любой вычислимой функции f найдётся такое число n, что $f(n) = d(n)$ (последнее равенство понимается в том смысле, что либо оба значения $f(n)$ и $d(n)$ не определены, либо оба определены и равны).*

Доказательство. По существу всё уже сказано: такова диагональная функция $d(n) = U(n, n)$ (здесь U — вычислимая функция двух аргументов, универсальная для класса вычислимых функций одного аргумента). Любая вычислимая функция f есть U_n при некотором n и потому $f(n) = U_n(n) = U(n, n) = d(n)$. □

Теорема 55. *Существует вычислимая функция, не имеющая всюду определённого вычислимого продолжения.*

Доказательство. Такова, например, функция $d'(n) = d(n) + 1$, где d — функция из предыдущей теоремы. В самом деле, любое её всюду определённое продолжение всюду отличается от d (в тех местах, где функция d определена, функция d' на единицу больше d и потому любое продолжение функции d' отличается от d; там, где d не определена, любая всюду определённая функция отличается от d). □

Задача 171. *Докажите, что и сама функция d из доказательства предыдущей теоремы не имеет вычислимого всюду определённого продолжения.*

§3. Перечислимое неразрешимое множество

Теперь мы можем доказать обещанное утверждение.

Теорема 56. *Существует перечислимое неразрешимое множество. (Переформулировка: существует перечислимое множество с неперечислимым дополнением.)*

Доказательство. Рассмотрим вычислимую функцию $f(x)$, не имеющую всюду определённого вычислимого продолжения. Её область определения F будет искомым множеством. В самом деле, F перечислимо (по одному из определений перечислимости). Если бы F было разрешимо, то функция

$$g(x) = \begin{cases} f(x), \text{ если } x \in F, \\ 0, \text{ если } x \notin F \end{cases}$$

была бы вычислимым всюду определённым продолжением функции f (при вычислении $g(x)$ мы сначала проверяем, лежит ли x в F, если лежит, то вычисляем $f(x)$). □

Полезно проследить, какое именно множество в итоге оказалось перечислимым и неразрешимым. Легко понять, что это множество тех n, при которых $U(n, n)$ определено. Если вспомнить конструкцию функции U, то это множество тех n, при которых n-я программа останавливается на n. Поэтому иногда говорят, что "проблема самоприменимости" (применимости программы к своему номеру) неразрешима.

Заметим, что отсюда следует, что и область определения всей универсальной функции U является перечислимым неразрешимым множеством пар. (Если бы проблема выяснения применимости программы к произвольному аргументу была бы разрешима, то и её частный случай — применимость программы к себе — был бы разрешим.)

Задача 172. *Пусть U — перечислимое множество пар натуральных чисел, универсальное для класса всех перечислимых множеств натуральных чисел. Докажите, что его "диагональное сечение" $K = \{x \mid \langle x, x\rangle \in U\}$ является перечислимым неразрешимым множеством.*

Задача 173. *Некоторое множество S натуральных чисел разрешимо. Разложим все числа из S на простые множители и составим множество D всех простых чисел, встречающихся в этих разложениях. Можно ли утверждать, что множество D разрешимо?*

Задача 174. *Множество $U \subset \mathbb{N} \times \mathbb{N}$ разрешимо. Можно ли утверждать, что множество "нижних точек" множества U, то есть множество*

$$V = \{\langle x, y\rangle \mid \ (\langle x, y\rangle \in U) \text{ и } (\langle x, z\rangle \notin U \text{ для всех } z < y)\}$$

является разрешимым? Можно ли утверждать, что V перечислимо, если U перечислимо?

Задача 175. *Покажите, что существуют перечислимые снизу, но не вычислимые числа в смысле определений, данных на с. 148. (Указание. Рассмотрим сумму ряда $\sum 2^{-k}$ по всем k из какого-либо перечислимого множества P. Она всегда перечислима снизу, но будет вычислимой только при разрешимом P.)*

Мы вернёмся к вычислимым действительным числам в задаче 179

ГЛАВА IX

Нумерации и операции

§1. Главные универсальные функции

Очевидно, композиция двух вычислимых функций вычислима. При этом это утверждение кажется "эффективным" в том смысле, что по программам двух функций можно алгоритмически получить программу их композиции. В разумном языке программирования она будет состоять из двух подпрограмм, соответствующих двум вычислимым функциям, и главной программы с единственной строкой "`return (f(g(x)))`".

Однако мы хотим говорить не о программах (чтобы не вдаваться в детали языка программирования), а о номерах функций. Для этого у нас есть средства. Именно, всякая универсальная функция U для класса вычислимых функций одного аргумента задаёт нумерацию этого класса: число n является номером функции $U_n\colon x \mapsto U(n,x)$.

Вообще *нумерацией* (более точно, *натуральной нумерацией*) произвольного множества $\mathcal{F}$ называют всюду определённое отображение $\nu\colon \mathbb{N} \to \mathcal{F}$, область значений которого есть всё множество $\mathcal{F}$. Если $\nu(n) = f$, то число n называют *номером* объекта f. Таким образом, всякая функция двух аргументов задаёт нумерацию некоторого класса функций одного аргумента (и является универсальной для этого класса).

Наша цель — сформулировать и доказать такое утверждение: (при некоторых условиях на нумерацию вычислимых функций) существует алгоритм, который по любым двум номерам вычислимых функций даёт некоторый номер их композиции.

Прежде всего мы потребуем, чтобы универсальная функция, задающая нумерацию, была вычислимой. (Такие нумерации называют *вычислимыми*.) Однако этого условия мало: нам потребуется, чтобы нумерация была, как говорят, *главной* (*гёделевой*).

Пусть U — двуместная вычислимая универсальная функция для класса одноместных вычислимых функций (термин "k-местная функция" означает "функция k аргументов"). Её называют *главной* универсальной функцией, если для любой двуместной вычислимой функции V существует всюду определённая вычислимая функция $s(m)$, для которой

$$V(m,x) = U(s(m),x)$$

при всех m и x (равенство понимается, как обычно, в том смысле, что либо

оба значения не определены, либо определены и равны).

Другими словами, $V_m = U_{s(m)}$, то есть функция s даёт по V-номеру некоторой функции некоторый U-номер той же функции.

Теорема 57. *Существует главная универсальная функция.*

Доказательство. (Первый способ.) Покажем, что описанное в доказательстве теоремы 51 (с. 149) построение универсальной функции даёт главную универсальную функцию. Напомним, что мы перечисляли все программы $p_0, p_1, p_2, \ldots$ какого-то естественного языка программирования в порядке возрастания их длин и полагали $U(n, x)$ равным результату применения программы p_n к входу x. Пусть теперь есть какая-то другая вычислимая функция V двух аргументов. Нам надо по любому натуральному m получить программу функции V_m, то есть функции, которая получится, если в V зафиксировать первый аргумент равным m. Ясно, что такую программу (в большинстве языков программирования) получить легко — надо только в программе для V заменить первый аргумент на определение константы (или использовать программу для V в качестве подпрограммы, а в основной программе вызывать V с фиксированным первым аргументом).

(Второй способ.) Но можно и не вдаваться в детали построения универсальной функции, а воспользоваться лишь фактом её существования.

Заметим сначала, что существует вычислимая функция трёх аргументов, универсальная для класса вычислимых функций двух аргументов, то есть такая функция T, что при фиксации первого аргумента среди функций $T_n(u, v) = T(n, u, v)$ встречаются все вычислимые функции двух аргументов.

Такую функцию можно построить так. Фиксируем некоторую вычислимую нумерацию пар, то есть вычислимое взаимно однозначное соответствие $\langle u, v\rangle \leftrightarrow [u, v]$ между $\mathbb{N} \times \mathbb{N}$ и $\mathbb{N}$; число $[u, v]$, соответствующее паре $\langle u, v\rangle$, мы будем называть номером этой пары. Если теперь R — двуместная вычислимая универсальная функция для вычислимых одноместных функций, то вычислимая функция T, определённая формулой $T(n, u, v) = R(n, [u, v])$, будет универсальной для вычислимых двуместных функций. В самом деле, пусть F — произвольная вычислимая функция двух аргументов. Рассмотрим вычислимую одноместную функцию f, определённую соотношением $f([u, v]) = F(u, v)$. Поскольку R универсальна, найдётся число n, для которого $R(n, x) = f(x)$ при всех x. Для этого n выполнены равенства $T(n, u, v) = R(n, [u, v]) = f([u, v]) = F(u, v)$, и потому n-ое сечение функции T совпадает с F. Итак, универсальная функция трёх аргументов построена.

Теперь используем её для определения главной универсальной функции U двух аргументов. Неформально говоря, мы встроим внутрь U все другие вычислимые функции двух аргументов, и тем самым U станет главной. Формально говоря, положим $U([n,u],v) = T(n,u,v)$ и проверим, что функция U будет главной. Любая вычислимая функция V двух аргументов встречается среди сечений функции T: можно найти такое n, что $V(u,v) = T(n,u,v)$ для всех u и v. Тогда $V(u,v) = U([n,u],v)$ для всех u и v и потому функция s, определённая формулой $s(u) = [n,u]$, удовлетворяет требованиям из определения главной универсальной функции. □

Нумерации, соответствующие главным универсальным функциям, называют *главными*, или *гёделевыми*.

Теперь уже можно доказать точный вариант утверждения, с которого мы начали.

Теорема 58. *Пусть U — двуместная главная универсальная функция для класса вычислимых функций одного аргумента. Тогда существует всюду определённая функция c, которая по номерам p и q двух функций одного аргумента даёт номер $c(p,q)$ их композиции: $U_{c(p,q)}$ есть композиция $U_p \circ U_q$, то есть*

$$U(c(p,q),x) = U(p,U(q,x))$$

для всех p, q и x.

Доказательство. Рассмотрим двуместную вычислимую функцию V, для которой $V([p,q],x) = U(p,U(q,x))$. По определению главной универсальной функции, найдётся такая всюду определённая одноместная вычислимая функция s, что $V(m,x) = U(s(m),x)$ для всех m и x. Тогда $V([p,q],x) = U(s([p,q]),x)$ и потому функция c, определённая соотношением $c(p,q) = s([p,q])$, будет искомой. □

Повторим это доказательство неформально. Определение главной универсальной функции требует, чтобы для любого другого языка программирования, для которого имеется вычислимый интерпретатор V, существовал бы вычислимый транслятор s программ этого языка в программы языка U. (Для краткости мы не различаем программы и их номера и рассматриваем число m как U-программу функции U_m.)

Теперь рассмотрим новый способ программирования, при котором пара $\langle p,q\rangle$ объявляется программой композиции функций с U-программами p и q. По условию, такие программы можно алгоритмически транслировать в U-программы, что и требовалось доказать.

Любопытно, что верно и обратное к теореме 58 утверждение:

Задача 176. *Пусть U — двуместная вычислимая универсальная функция для класса вычислимых функций одного аргумента. Если существует всюду определённая функция, которая по номерам p и q двух функций одного аргумента даёт какой-либо номер их композиции, то функция U является главной. (Указание: покажите, что по k можно алгоритмически получать U-номер функции $x \mapsto [k, x]$.)*

Естественный вопрос: существуют ли вычислимые универсальные функции, не являющиеся главными? Мы увидим дальше, что существуют.

Задача 177. *Изменим определение главной универсальной функции и будем требовать существования "транслятора" s лишь для универсальных вычислимых функций V (а не для любых, как раньше). Покажите, что новое определение эквивалентно старому. (Указание: любую функцию можно искусственно переделать в универсальную, "растворив" в ней любую другую универсальную функцию.)*

Задача 178. *Пусть U — главная универсальная функция. Докажите, что для любой вычислимой функции $V(m, n, x)$ существует такая всюду определённая вычислимая функция $s(m, n)$, что $V(m, n, x) = U(s(m, n), x)$ при всех m, n и x. (Указание: объединить m и n в пару.)*

§2. Вычислимые последовательности вычислимых функций

Пусть дана некоторая последовательность $f_0, f_1, \ldots$ вычислимых функций одного аргумента. Мы хотим придать смысл выражению "последовательность $i \mapsto f_i$ вычислима". Это можно сделать двумя способами:

- можно называть эту последовательность вычислимой, если функция F двух аргументов, заданная формулой $F(i, n) = f_i(n)$, является вычислимой.
- можно называть эту последовательность вычислимой, если существует вычислимая последовательность чисел $c_0, c_1, \ldots$, для которой c_i является одним из номеров функции f_i.

Второе определение (в отличие от первого) зависит от выбора нумерации.

Теорема 59. *Если нумерация является вычислимой (то есть соответствующая универсальная функция вычислима), то из второго определения следует первое. Если нумерация является к тому же главной, то из первого определения следует второе.*

(Впредь, говоря о вычислимой последовательности вычислимых функций, мы будем всегда предполагать, что нумерация является главной, так что можно пользоваться любым из двух определений.)

Доказательство. Если U — вычислимая универсальная функция, а последовательность $i \mapsto c_i$ вычислима, то функция $F\colon \langle i, x\rangle \mapsto f_i(x) = U(c_i, x)$ вычислима как результат подстановки одной вычислимой функции в другую.

Напротив, если функция F вычислима, а универсальная функция U является главной, то функция-транслятор, существующая по определению главной универсальной функции, как раз и даёт по i один из номеров функции f_i. □

Задача 179. *Пусть фиксирована главная универсальная функция для класса вычислимых функций одного аргумента. Тогда возникает нумерация вычислимых действительных чисел в соответствии с определением на с. 148: номером числа α является любой номер любой функции, которая по рациональному $\varepsilon > 0$ даёт ε-приближение к α.*

*(**а**) Покажите, что существует алгоритм, который по любым двум номерам двух вычислимых действительных чисел даёт (некоторый) номер их суммы.*

*(**б**) Покажите, что не существует алгоритма, который по любому номеру любого вычислимого действительного числа отвечает на вопрос, равно ли это число нулю.*

*(**в**) Как мы видели в задаче 166, всякое вычислимое действительное число имеет вычислимое десятичное разложение. Покажите, что тем не менее нет алгоритма, который по любому номеру любого вычислимого действительного числа даёт номер вычислимой функции, задающей его десятичное разложение.*

§3. Главные универсальные множества

По аналогии с функциями, перечислимое множество $W \subset \mathbb{N} \times \mathbb{N}$ называется *главным универсальным перечислимым множеством* (для класса всех перечислимых подмножеств $\mathbb{N}$), если для любого другого перечислимого множества $V \subset \mathbb{N} \times \mathbb{N}$ найдётся такая всюду определённая вычислимая функция $s\colon \mathbb{N} \to \mathbb{N}$, что

$$\langle n, x\rangle \in V \Leftrightarrow \langle s(n), x\rangle \in W$$

для всех n и x. (Очевидно, что из этого свойства следует универсальность.)

Как и для функций, можно перейти к нумерациям. Каждое множество $U \subset \mathbb{N} \times \mathbb{N}$ задаёт нумерацию некоторого семейства подмножеств натурального ряда: число n является номером n-го сечения $U_n = \{x \mid \langle n, x\rangle \in U\}$. Перечислимое подмножество множества $\mathbb{N} \times \mathbb{N}$ задаёт нумерацию некоторого семейства перечислимых подмножеств натурального ряда; такие нумерации называют *вычислимыми*. Перечислимое множество $W \subset \mathbb{N} \times \mathbb{N}$ универсально, если и только если всякое перечислимое подмножество натурального ряда имеет W-номер; оно является главным тогда и только тогда, когда любая вычислимая нумерация V (любого семейства перечислимых множеств) вычислимо сводится к W-нумерации в том смысле, что $V_n = = W_{s(n)}$ для некоторой вычислимой функции s и для всех n.

Теорема 60. *Существует главное универсальное перечислимое множество $W \subset \mathbb{N} \times \mathbb{N}$.*

Доказательство. Эта теорема является очевидным следствием такого утверждения:

Лемма. Область определения главной универсальной функции для класса вычислимых функций одного аргумента является главным универсальным множеством для класса перечислимых подмножеств $\mathbb{N}$.

Доказательство леммы. Пусть U — главная универсальная функция, а W — область её определения. Пусть $V \subset \mathbb{N} \times \mathbb{N}$ — произвольное перечислимое множество. Рассмотрим вычислимую функцию G с областью определения V. Поскольку функция U является главной, найдётся всюду определённая вычислимая функция $s\colon \mathbb{N} \to \mathbb{N}$, для которой $G_n = U_{s(n)}$ при всех n. Тогда равны и области определения функций G_n и $U_{s(n)}$, то есть $V_n = W_{s(n)}$. □

Задача 180. *Постройте главное универсальное множество непосредственно, используя универсальное подмножество $\mathbb{N}^3$ (по аналогии с выше приведённым построением главной универсальной функции).*

Как и для функций, мы теперь можем доказывать, что различным операциям над множествами соответствуют вычислимые преобразования номеров. Вот лишь один пример такого рода:

Теорема 61. *Пусть $W \subset \mathbb{N} \times \mathbb{N}$ — главное универсальное перечислимое множество. Тогда по W-номерам двух перечислимых множеств можно алгоритмически получить номер их пересечения: существует такая вычислимая всюду определённая функция двух аргументов s, что*

$$W_{s(m,n)} = W_m \cap W_n$$

для любых двух m и n.

Доказательство. Рассмотрим множество $V \subset \mathbb{N} \times \mathbb{N}$, определённое так:

$$\langle [m,n], x\rangle \in V \Leftrightarrow x \in (W_m \cap W_n)$$

(здесь квадратные скобки обозначают номер пары) и применим к нему определение главного универсального множества. □

Как и для функций, понятие вычислимости последовательности перечислимых множеств может быть определено двояко: можно считать вычислимой последовательность $V_0, V_1, \ldots$ сечений произвольного перечислимого множества V, а можно требовать, чтобы по i можно было алгоритмически указать один из номеров i-го члена последовательности в главной нумерации. Эти определения равносильны (доказательство полностью аналогично рассуждению для функций).

§4. Множества номеров

Начнём с такого примера. Рассмотрим множество номеров нигде не определённой функции для какой-либо главной нумерации. Будет ли оно разрешимо? Другими словами, можно ли по номеру функции в главной нумерации определить, является ли эта функция нигде не определённой?

Прежде чем отвечать на этот вопрос, заметим, что ответ не зависит от того, какая главная нумерация выбрана. В самом деле, если есть две разные главные нумерации, то они, как говорят, "сводятся" друг к другу: по номеру функции в одной нумерации можно алгоритмически получить номер той же функции в другой нумерации. Если бы в одной нумерации можно было бы проверять "нигде-не-определённость" функции, то это можно было бы делать и в другой (применив "функции перехода").

Следующая теорема показывает, что ответ на исходный вопрос будет отрицательным.

Теорема 62. *Пусть U — произвольная главная универсальная функция. Тогда множество тех n, при которых функция U_n является нигде не определённой, неразрешимо.*

Доказательство. Используем метод, называемый "сведéнием" — покажем, что если бы это множество было разрешимым, то и вообще любое перечислимое множество было бы разрешимым. (Что, как мы знаем, неверно.)

Пусть K — произвольное перечислимое неразрешимое множество. Рассмотрим такую вычислимую функцию V двух аргументов:

$$V(n,x) = \begin{cases} 0, \text{ если } n \in K, \\ \text{не определено, если } n \notin K. \end{cases}$$

Как видно, второй аргумент этой функции фиктивен, и она по существу совпадает с полухарактеристической функцией множества K от первого аргумента. Очевидно, эта функция имеет сечения двух типов: при $n \in K$ сечение V_n является нулевой функцией, при $n \notin K$ — нигде не определённой функцией.

Так как функция U является главной, существует вычислимая всюду определённая функция s, для которой $V(n,x) = U(s(n),x)$ при всех n и x, т. е. $V_n = U_{s(n)}$. Поэтому при $n \in K$ значение $s(n)$ является U-номером нулевой функции, а при $n \notin K$ значение $s(n)$ является U-номером нигде не определённой функции. Поэтому если бы множество U-номеров нигде не определённой функции разрешалось бы некоторым алгоритмом, то мы бы могли применить этот алгоритм к $s(n)$ и узнать, принадлежит ли число n множеству K или нет. Таким образом, множество K было бы разрешимым в противоречии с нашим предположением. □

В частности, мы можем заключить, что нигде не определённая функция имеет бесконечно много номеров в любой главной нумерации (поскольку любое конечное множество разрешимо).

Кроме того, можно заметить, что множество номеров нигде не определённой функции не только не разрешимо, но и не перечислимо. В самом деле, его дополнение — множество всех номеров всех функций с непустой областью определения — перечислимо. (Это верно для любой вычислимой нумерации, а не только для главной: параллельно вычисляя $U(n,x)$ для всех n и x, мы можем печатать те n, для которых обнаружилось x, при котором $U(n,x)$ определено.) А если дополнение неразрешимого множества перечислимо, то само множество неперечислимо (по теореме Поста, с. 145).

Справедливо и более общее утверждение, называемое иногда теоремой Успенского – Райса. Обозначим класс всех вычислимых функций (одного аргумента) через $\mathcal{F}$.

Теорема 63. *Пусть $\mathcal{A} \subset \mathcal{F}$ — произвольное нетривиальное свойство вычислимых функций (нетривиальность означает, что есть как функции, ему удовлетворяющие, так и функции, ему не удовлетворяющие, то есть что множество $\mathcal{A}$ непусто и не совпадает со всем $\mathcal{F}$). Пусть U — главная универсальная функция. Тогда не существует алгоритма, кото-*

рый по U-номеру вычислимой функции проверял бы, обладает ли она свойством $\mathcal{A}$. Другими словами, множество $\{n \mid U_n \in \mathcal{A}\}$ неразрешимо.

Доказательство. Посмотрим, принадлежит ли нигде не определённая функция (обозначим её ζ) классу $\mathcal{A}$, и возьмём произвольную функцию ξ "с другой стороны" (если $\zeta \in \mathcal{A}$, то $\xi \notin \mathcal{A}$ и наоборот).

Далее действуем как раньше, но только вместо нулевой функции возьмём функцию ξ: положим

$$V(n,x) = \begin{cases} \xi(x), \text{ если } n \in K, \\ \text{не определено, если } n \notin K. \end{cases}$$

Как и раньше, функция V будет вычислимой (для данных n и x мы ожидаем появления n в множестве K, после чего вычисляем $\xi(x)$). При $n \in K$ функция V_n совпадает с ξ, при $n \notin K$ — с ζ. Таким образом, проверяя свойство $V_n \in \mathcal{A}$ (если бы это можно было сделать вопреки утверждению теоремы), можно было бы узнать, принадлежит ли число n множеству K или нет. □

Некоторым недостатком этого доказательства является его несимметричность (с одной стороны от $\mathcal{A}$ мы берём нигде не определённую функцию, с другой стороны — любую). Вот более симметричный вариант. Покажем, что если свойство $\mathcal{A}$ можно распознавать по U-номерам, то любые два непересекающихся перечислимых множества P и Q отделимы разрешимым множеством. Выберем какие-нибудь две функции ξ и η, находящиеся "по разные стороны" от $\mathcal{A}$. Рассмотрим функцию

$$V(n,x) = \begin{cases} \xi(x), \text{ если } n \in P, \\ \eta(x), \text{ если } n \in Q, \\ \text{не определено, если } n \notin P \cup Q. \end{cases}$$

Эта функция вычислима: для заданных n и x ожидаем, пока n появится либо в P, либо в Q, после чего запускаем вычисление соответственно $\xi(x)$ или $\eta(x)$.

Если $n \in P$, то V_n совпадает с ξ; если $n \in Q$, то V_n совпадает с η. Поэтому, проверяя, принадлежит ли V_n классу $\mathcal{A}$, мы могли бы разрешимо отделить P от Q. Получаем противоречие, которое и завершает этот более симметричный вариант доказательства.

Второй вариант доказательства показывает, что верно следующее усиление этой теоремы: для любых различных вычислимых функций φ и ψ и любой главной универсальной функции U множества всех U-номеров функции φ и функции ψ не отделимы разрешимым множеством. (Заметим, что эти множества не перечислимы, как мы впоследствии увидим.)

Теперь легко указать пример вычислимой универсальной функции, не являющейся главной. Достаточно сделать так, чтобы нигде не определённая функция имела единственный номер. Это несложно. Пусть $U(n,x)$ — произвольная вычислимая универсальная функция. Рассмотрим множество D всех U-номеров всех функций с непустой областью определения. Как мы уже говорили, это множество перечислимо. Рассмотрим всюду определённую вычислимую функцию d, его перечисляющую: $D = \{d(0), d(1), \dots\}$. Теперь рассмотрим функцию $V(i,x)$, для которой $V(0,x)$ не определено ни при каком x, а $V(i{+}1,x) = U(d(i),x)$. Другими словами, функция V_0 нигде не определена, а функция V_{i+1} совпадает с $U_{d(i)}$. Легко понять, что функция V вычислима; она универсальна по построению, и единственным V-номером нигде не определённой функции является число 0.

На самом деле существуют и более экзотические нумерации: как показал Фридберг, можно построить универсальную вычислимую функцию, для которой каждая вычислимая функция будет иметь ровно один номер. Соответствующие нумерации называют *однозначными*; очевидно, они не могут быть главными. Забавная переформулировка: можно разработать такой язык программирования, в котором каждую программистскую задачу можно решить единственным образом. (Доказательство этой теоремы трудно и не приводится, на русском языке оно есть в книжке А. И. Мальцева "Алгоритмы и рекурсивные функции.)

Аналогичное утверждение верно и для нумераций перечислимых множеств.

ГЛАВА X

Теорема о неподвижной точке

§1. Неподвижная точка и отношения эквивалентности

Теорема 64. *Пусть U — главная вычислимая универсальная функция для класса вычислимых функций одного аргумента, а h — произвольная всюду определённая вычислимая функция одного аргумента. Тогда существует такое число n, что $U_n = U_{h(n)}$, то есть n и $h(n)$ — номера одной функции.*

Другими словами, нельзя найти алгоритма, преобразующего программы, который бы по каждой программе давал другую (не эквивалентную ей). Эту теорему называют *теоремой Клини о неподвижной точке* или *теоремой о рекурсии.*

Доказательство. Мы будем действовать по аналогии с построением вычислимой функции, не имеющей всюду определённого вычислимого продолжения (глава VIII).

Рассмотрим произвольное отношение эквивалентности (которое мы будем обозначать $x \equiv y$) на множестве натуральных чисел. Мы покажем, что следующие два свойства этого отношения не могут выполняться одновременно:

- Для всякой вычислимой функции f существует всюду определённая вычислимая функция g, являющаяся её $\equiv$-продолжением (это означает, что если $f(x)$ определено при некотором x, то $g(x) \equiv f(x)$).
- Существует всюду определённая вычислимая функция h, не имеющая $\equiv$-неподвижной точки (то есть функция, для которой $n \not\equiv h(n)$ для всех n).

Если $x \equiv y$ — отношение равенства ($x = y$), то второе свойство выполнено (положим, например, $h(n) = n + 1$), поэтому не выполнено первое. Теорема о неподвижной точке получится, если $x \equiv y \longleftrightarrow U_x = U_y$ (x и y — номера одной и той же функции). В этом случае выполнено первое свойство, как мы сейчас убедимся, и потому не выполнено второе.

Почему выполнено первое свойство? Пусть f — произвольная вычислимая функция одного аргумента. Рассмотрим функцию $V(n, x) = U(f(n), x)$. Поскольку U является главной универсальной функцией, найдётся всюду определённая функция s, для которой $V(n, x) = U(s(n), x)$ при всех n

и x. Эта функция и будет искомым $\equiv$-продолжением. В самом деле, если $f(n)$ определено, то $s(n)$ будет другим номером той же функции, что и $f(n)$. (Отметим, что если $f(n)$ не определено, то $s(n)$ будет одним из номеров нигде не определённой функции.)

Для завершения доказательства теоремы о неподвижной точке осталось проверить, что указанные два свойства отношения эквивалентности несовместны. Это делается так же, как в теореме 54 (раздел 2). Возьмём вычислимую функцию f, от которой никакая вычислимая функция не может отличаться всюду (например, диагональную функцию $x \mapsto U(x,x)$ для некоторой вычислимой универсальной функции U). По предположению существует всюду определённое вычислимое $\equiv$-продолжение g функции f. Рассмотрим функцию $t(x) = h(g(x))$, где h — вычислимая всюду определённая функция, не имеющая $\equiv$-неподвижной точки. Тогда t будет всюду отличаться от f. В самом деле, если $f(x)$ определено, то $f(x) \equiv g(x) \not\equiv h(g(x)) = t(x)$, и потому $f(x) \neq t(x)$. Если же $f(x)$ не определено, то этот факт сам по себе уже отличает $f(x)$ и $t(x)$. □

Теорему о неподвижной точке можно переформулировать и так:

Теорема 65. *Пусть $U(n,x)$ — главная вычислимая универсальная функция для класса вычислимых функций одного аргумента. Пусть $V(n,x)$ — произвольная вычислимая функция. Тогда функции U и V совпадают на некотором сечении: найдётся такое p, что $U_p = V_p$, то есть $U(p,n) = V(p,n)$ для любого n.*

Доказательство. Так как функция U является главной, найдём такую всюду определённую вычислимую функцию h, что $V(n,x) = U(h(n),x)$ при всех n и x. Осталось взять в качестве p неподвижную точку функции h. □

(Пример следствия из этой теоремы: как бы ни старались разработчики, для любых двух версий компилятора существует программа, которая одинаково работает в обеих версиях — например, зацикливается и там, и там. Впрочем, это всё же не наверняка, а только если компилятор задаёт главную универсальную функцию — но надо очень постараться, чтобы это было не так!)

Поучительно развернуть цепочку приведённых рассуждений и проследить, как строится неподвижная точка. Для наглядности вместо $U(n,x)$ мы будем писать $[n](x)$ и читать это "результат применения программы n к входу x";

Рассуждение начинается с рассмотрения "диагональной" функции $U(x,x)$, которую теперь можно записать как $[x](x)$ (результат применения программы x к себе). Далее мы строим её всюду определённое

$\equiv$-продолжение. Это делается так. Выражение $[[x](x)](y)$ вычислимо зависит от двух аргументов. Мы вспоминаем, что U есть главная универсальная функция, и находим такую программу g, что $[[g](x)](y) = [[x](x)](y)$ при всех x и y. При этом $[g](x)$ определено для всех x. Пусть мы хотим найти неподвижную точку программы h. Мы рассматриваем композицию $[h]([g](x))$. Это выражение вычислимо зависит от x, и потому существует программа t, для которой $[t](x) = [h]([g](x))$ при всех x. Эта программа применима ко всем x, поскольку таковы h и g. Теперь неподвижной точкой будет $[g](t)$. Чтобы убедиться в этом, мы должны проверить, что $[[g](t)](x) = [[h]([g](t))](x)$ для всех x. В самом деле, по свойству g имеем $[[g](t)](x) = [[t](t)](x)$. Вспоминая определение t, это выражение можно переписать как $[[h]([g](t))](x)$ — что как раз и требовалось.

§2. Программа, печатающая свой текст

Классическим примером применения теоремы о неподвижной точке является такое её следствие: существует программа, печатающая (на любом входе) свой собственный текст. В самом деле, если бы такой программы не было, то преобразование

$$p \mapsto (\text{программа, которая на любом входе печатает } p)$$

не имело бы неподвижной точки.

Формально говоря, это следствие можно выразить так:

Теорема 66. *Пусть $U(n, x)$ — главная вычислимая универсальная функция для класса всех вычислимых функций одного аргумента. Тогда существует такое число p, что $U(p, x) = p$ для любого x.*

В программистских терминах: пусть $U(p, x)$ — результат применения C-программы p к стандартному входу x. (Уточнения: (1) мы отождествляем числа и последовательности байтов; (2) если программа не завершает работы, мы считаем, что результат не определён, даже если на стандартный выход что-то послано.) Ясно, что функция U будет главной универсальной функцией. Поэтому к ней можно применить сформулированное только что утверждение; получим программу p, которая при любом входе на выходе даёт p.

Ясно, что это рассуждение применимо для любого языка программирования; то, что мы упомянули язык C, роли не играет.

Задача 181. *Докажите, что существует C-программа, которая печатает свой текст задом наперёд.*

Задача 182. *Покажите, что есть две различные С-программы P и Q с такими свойствами: программа P печатает текст программы Q, а программа Q печатает текст программы P. (Если не требовать различия между P и Q, то в качестве P и Q можно взять одну и ту же программу, печатающую свой текст.)*

Задача 183. *Напишите программу на языке С, печатающую свой текст (без использования функций работы с файлами наподобие* `read`*).*

На русском языке подобная программа имела бы вид: напечатать два раза, второй раз в кавычках, такой текст: "напечатать два раза, второй раз в кавычках, такой текст:".

Задача 184. *Напишите программу на языке С, печатающую свой текст задом наперёд.*

Сделав ещё один шаг, можно получить и доказательство теоремы о неподвижной точке. Пусть h — некоторое преобразование С-программ, у которого мы хотим найти неподвижную точку. Тогда напишем программу наподобие печатающей себя, которая будет записывать свой текст в строку `p`, затем применять h к `p`, получая некоторую другую строку `q`, а затем запускать интерпретатор языка С на строке `q` (используя в качестве входа программы `q` вход исходной программы). Конечно, эта программа уже не будет такой короткой, так как будет включать в себя (и даже два раза — первый раз просто так, а второй раз в кавычках) интерпретатор С, написанный на С.

Ясно, что такая программа будет неподвижной точкой преобразования h, так как её выполнение начинается ровно с того, что вычисляется значение функции h на её тексте, после чего это значение воспринимается как программа и применяется к входу.

На самом деле это доказательство в сущности повторяет предыдущее, только в более программистских терминах.

§3. Несколько замечаний

3.1. Бесконечное множество неподвижных точек

Теорема 64 (о неподвижной точке) утверждает существование хотя бы одной неподвижной точки. Легко понять, что на самом деле их бесконечно много: в обозначениях этой теоремы существует бесконечно много чисел n, при которых $U_n = U_{h(n)}$

Это можно объяснить, например, так: если бы неподвижных точек было бы конечное число, то можно было бы изменить функцию h в этих точках так, чтобы неподвижных точек не осталось. Недостаток этого рассуждения в том, что оно не позволяет эффективно перечислять неподвижные точки (указать для данной функции h бесконечное перечислимое множество, состоящее из её неподвижных точек). Можно сделать и это, если вспомнить доказательство теоремы 64. В нём неподвижными точками оказывались числа $[g](t)$, а функцию g можно выбрать так, чтобы все её значения были больше любого наперёд заданного числа

Задача 185. *Проведите это рассуждение подробно.*

3.2. Неподвижная точка с параметром

Если преобразователь программ вычислимо зависит от некоторого параметра, то и неподвижную точку можно выбрать вычислимо зависящей этого параметра. Точный смысл этого утверждения таков:

Теорема 67. *Пусть U — главная универсальная функция для класса вычислимых функций одного аргумента, а h — всюду определённая вычислимая функция двух аргументов. Тогда существует всюду определённая вычислимая функция n одного аргумента, которая по любому p указывает неподвижную точку для функции h_p, так что $U_{h(p,n(p))} = U_{n(p)}$, или, другими словами,*

$$U(h(p, n(p)), x) = U(n(p), x)$$

при всех p и x (как обычно, обе части могут быть одновременно не определены).

Доказательство. Мы видели, что неподвижная точка строится конструктивно. Поэтому если мы ищем неподвижную точку для функции h_p, вычислимо зависящей от параметра p, то и результат нашего построения будет вычислимо зависеть от параметра p.

Конечно, можно было бы формально записать рассуждение, реализующее этот план, но оно довольно громоздко (и вряд ли от этого доказательство станет более понятным). □

В этой теореме мы предполагали, что семейство функций h_p состоит из всюду определённых функций. На самом деле это не обязательно: для произвольного вычислимого семейства вычислимых функций h_p (другими словами, для произвольной вычислимой функции h двух аргументов) существует всюду определённая вычислимая функция n одного аргумента с

таким свойством: при каждом p либо функция h_p не определена в точке $n(p)$, либо $n(p)$ является неподвижной точкой функции h_p.

Задача 186. *Убедитесь, что приведённая в доказательстве теоремы 67 конструкция как раз и даёт функцию $n(p)$ с таким свойством.*

Задача 187. *Объединяя сделанные выше замечания, покажите, что по любой вычислимой функции h (заданной своим номером относительно фиксированной главной универсальной функции) можно эффективно указать бесконечно много чисел, каждое из которых либо будет неподвижной точкой для функции h, либо точкой, где эта функция не определена.*

3.3. Неподвижная точка для перечислимых множеств

Всё сказанное почти без изменений переносится на главные нумерации перечислимых множеств (если W — главное универсальное перечислимое множество, то всякая вычислимая всюду определённая функция h имеет неподвижную точку n, для которой $W_n = W_{h(n)}$).

В самом деле, если W — главное универсальное перечислимое множество, то к отношению эквивалентности

$$a \equiv b \Leftrightarrow W_a = W_b$$

применимо рассуждение из доказательства теоремы 64, поскольку любая вычислимая функция f имеет вычислимое всюду определённое $\equiv$-продолжение.

Проверим это. Для этого рассмотрим множество

$$V = \{\langle p, x\rangle \mid f(p) \text{ определено и } \langle f(p), x\rangle \in W\}.$$

Легко понять, что это множество перечислимо (например, оно есть область определения вычислимой функции $\langle p, x\rangle \mapsto w(f(p), x)$, где w — вычислимая функция с областью определения W). При этом $V_p = W_{f(p)}$, если $f(p)$ определено, и $V_p = \varnothing$, если $f(p)$ не определено. Вспоминая, что W является главным универсальным множеством, мы находим всюду определённую функцию s, для которой $V_p = W_{s(p)}$. Таким образом, $W_{s(p)} = W_{f(p)}$ для тех p, для которых $f(p)$ определено, что и требовалось.

Задача 188. *Пусть W — главное универсальное множество (для класса всех перечислимых подмножеств натурального ряда).* **(а)** *Покажите, что найдётся число x, для которого $W_x = \{x\}$.* **(б)** *Покажите, что найдутся различные числа x и y, для которых $W_x = \{y\}$ и $W_y = \{x\}$.*

3.4. Пример использования

Простейшее (хотя не очень типичное) применение теоремы о неподвижной точке — ещё одно доказательство теоремы 63 о неразрешимости свойств вычислимых функций. В самом деле, пусть есть нетривиальное свойство $\mathcal{A}$ вычислимых функций, которое можно распознавать по номерам функций в главной нумерации U. Пусть p — какой-то номер какой-то функции U_p, обладающей этим свойством, а q — какой-то номер какой-то функции U_q, им не обладающим. Тогда функция

$$h(x) = \begin{cases} q, \text{ если функция } U_x \text{ обладает свойством } \mathcal{A}, \\ p, \text{ если функция } U_x \text{ не обладает свойством } \mathcal{A} \end{cases}$$

будет вычислимой и не будет иметь неподвижной точки.

ГЛАВА XI

Машины Тьюринга

§1. Зачем нужны простые вычислительные модели?

До сих пор нам было удобно ссылаться на программистский опыт, говоря об алгоритмах, программах, интерпретаторах, пошаговом выполнении и т. д. Это позволяло нам игнорировать детали построения тех или иных алгоритмов под тем предлогом, что читатель их легко восстановит (или хотя бы поверит — всё-таки не каждый читатель в своей жизни писал интерпретатор языка C на C).

Но в некоторых случаях этого недостаточно. Пусть, например, мы хотим доказать алгоритмическую неразрешимость какой-то задачи, в определении которой ничего не говорится о программах. Это обычно делается так. Мы показываем, что проблема остановки сводится к этой задаче. Для этого мы моделируем работу произвольного алгоритма в терминах рассматриваемой задачи (что это значит, будет видно из приводимого ниже примера). При этом нам важно, чтобы определение алгоритма было как можно проще.

Таким образом, наш план таков. Мы опишем довольно просто определяемый класс машин (его можно выбирать по-разному, мы будем использовать так называемые машины Тьюринга), затем объявим, что всякая вычислимая функция может быть вычислена на такой машине.

Другая причина, по которой важны простые вычислительные модели (таких моделей много — разные виды машин Тьюринга, адресные машины и т. п.), связана с теорией сложности вычислений, когда нас начинает интересовать время выполнения программ. Но этот вопрос выходит за рамки классической теории алгоритмов.

§2. Машины Тьюринга: определение

Машина Тьюринга имеет бесконечную в обе стороны *ленту*, разделённую на квадратики (*ячейки*). В каждой ячейке может быть записан некоторый символ из фиксированного (для данной машины) конечного множества, называемого *алфавитом* данной машины. Один из символов алфавита выделен и называется "пробелом" — предполагается, что изначально вся лента пуста, то есть заполнена пробелами.

Машина Тьюринга может менять содержимое ленты с помощью специ-

альной читающей и пишущей *головки*, которая движется вдоль ленты. В каждый момент головка находится в одной из ячеек. Машина Тьюринга получает от головки информацию о том, какой символ та видит, и в зависимости от этого (и от своего внутреннего состояния) решает, что делать, то есть какой символ записать в текущей ячейке и куда сдвинуться после этого (налево, направо или остаться на месте). При этом также меняется внутреннее состояние машины (мы предполагаем, что машина — не считая ленты — имеет конечную память, то есть конечное число внутренних состояний). Ещё надо договориться, с чего мы начинаем и когда кончаем работу.

Таким образом, чтобы задать машину Тьюринга, надо указать следующие объекты:

- произвольное конечное множество A (*алфавит*); его элементы называются *символами*;
- некоторый выделенный символ $a_0 \in A$ (*пробел*, или *пустой символ*);
- конечное множество S, называемое множеством *состояний*;
- некоторое выделенное состояние $s_0 \in S$, называемое *начальным*;
- *таблицу переходов*, которая определяет поведение машины в зависимости от состояния и текущего символа (см. ниже);
- некоторое подмножество $F \subset S$, элементы которого называются *заключительными состояниями* (попав в такое состояние, машина останавливается).

Таблица переходов устроена следующим образом: для каждой пары ⟨текущее состояние, текущий символ⟩ указана тройка ⟨новое состояние, новый символ, сдвиг⟩. Здесь сдвиг — одно из чисел -1 (влево), 0 (на месте) и 1 (направо). Таким образом, таблица переходов есть функция типа $S \times A \to S \times A \times \{-1, 0, 1\}$, определённая на тех парах, в которых состояние не является заключительным.

Остаётся описать поведение машины Тьюринга. В каждый момент имеется некоторая *конфигурация*, складывающаяся из содержимого ленты (формально говоря, содержимое ленты есть произвольное отображение $\mathbb{Z} \to A$), текущей позиции головки (некоторое целое число) и текущего состояния машины (элемент S). Преобразование конфигурации в следующую происходит по естественным правилам: мы смотрим в таблице, что надо делать для данного состояния и для данного символа, то есть выясняем новое состояние машины, меняем символ на указанный и после этого сдвигаем головку влево, вправо или оставляем на месте. При этом, если новое состояние является одним из заключительных, работа машины заканчивается. Остаётся договориться, как мы подаём информацию на вход машины и что

считается результатом её работы. Будем считать, что алфавит машины, помимо пробела, содержит символы 0 и 1 (а также, возможно, ещё какие-то символы). Входом и выходом машины будут конечные последовательности нулей и единиц (двоичные слова). Входное слово записывается на пустой ленте, головка машины ставится в его первую клетку, машина приводится в начальное состояние и запускается. Если машина останавливается, результатом считается двоичное слово, которое можно прочесть, начиная с позиции головки и двигаясь направо (пока не появится символ, отличный от 0 и 1).

Таким образом, любая машина Тьюринга задаёт некоторую частичную функцию на двоичных словах. Все такие функции естественно назвать *вычислимыми на машинах Тьюринга.*

§3. Машины Тьюринга: обсуждение

Разумеется, наше определение содержит много конкретных деталей, которые можно было бы изменить. Например, лента может быть бесконечной только в одну сторону. Можно придать машине две ленты. Можно считать, что машина может либо написать новый символ, либо сдвинуться, но не то и другое вместе. Можно ограничить алфавит, считая, скажем, что в нём должно быть ровно 10 символов. Можно потребовать, чтобы в конце на ленте ничего не было, кроме результата работы (остальные клетки должны быть пусты). Все перечисленные и многие другие изменения не меняют класса вычислимых на машинах Тьюринга функций. Конечно, есть и небезобидные изменения. Например, если запретить машине двигаться налево, то это радикально поменяет дело — по существу лента станет бесполезной, так как к старым записям уже нельзя будет вернуться.

Как понять, какие изменения безобидны, а какие нет? Видимо, тут необходим некоторый опыт практического программирования на машинах Тьюринга, хотя бы небольшой. После этого уже можно представлять себе возможности машины, не выписывая программы полностью, а руководствуясь лишь приблизительным описанием. В качестве примера опишем машину, которая удваивает входное слово (изготавливает слово XX, если на входе было слово X).

Если машина видит пробел (входное слово пусто), она кончает работу. Если нет, она запоминает текущий символ и ставит пометку (в алфавите помимо символов 0 и 1 будут ещё их "помеченные варианты" $\overline{0}$ и $\overline{1}$). Затем она движется направо до пустой клетки, после чего пишет там копию запомненного символа. Затем она движется налево до пометки; уткнувшись в

пометку, отходит назад и запоминает следующий символ и так далее, пока не скопирует всё слово.

Имея некоторый опыт, можно за всеми этими фразами видеть конкретные куски программы для машины Тьюринга. Например, слова "запоминает символ и движется направо" означают, что есть две группы состояний, одна для ситуации, когда запомнен нуль, другая — когда запомнена единица, и внутри каждой группы запрограммировано движение направо до первой пустой клетки.

Имея ещё чуть больше опыта, можно понять, что в этом описании есть ошибка — не предусмотрен механизм остановки, когда всё слово будет скопировано, поскольку копии символов ничем не отличаются от символов исходного слова. Ясно и то, как ошибку исправить — надо в качестве копий писать специальные символы $\tilde{0}$ и $\tilde{1}$, а на последнем этапе все пометки удалить.

Задача 189. *Покажите, что функция "обращение", переворачивающая слово задом наперёд, вычислима на машине Тьюринга.*

Другой пример неформального рассуждения: объясним, почему можно не использовать дополнительных символов, кроме 0, 1 и пустого символа. Пусть есть машина с большим алфавитом из N символов. Построим новую машину, которая будет моделировать работу старой, но каждой клетке старой будет соответствовать блок из k клеток новой. Размер блока (число k) будет фиксирован так, чтобы внутри блока можно было бы закодировать нулями и единицами все символы большого алфавита. Исходные символы 0, 1 и пустой будем кодировать как 0, за которым идут $(k-1)$ пустых символов, 1, за которым идут $(k-1)$ пустых символов, и группу из k пустых символов. Для начала надо раздвинуть буквы входного слова на расстояние k, что можно сделать без дополнительных символов (дойдя до крайней буквы, отодвигаем её, затем дойдя до следующей, отодвигаем её и крайнюю и так далее); надо только понимать, что можно идентифицировать конец слова как позицию, за которой следует более k пустых символов. Ясно, что в этом процессе мы должны хранить в памяти некоторый конечный объём информации, так что это возможно. После этого уже можно моделировать работу исходной машины по шагам, и для этого тоже достаточно конечной памяти (т. е. конечного числа состояний), так как нам важна только небольшая окрестность головки моделируемой машины. Наконец, надо сжать результат обратно.

Утверждение о том, что всякая вычислимая функция вычислима на машине Тьюринга, называют *тезисом Тьюринга*. Конечно, его смысл зависит от того, что понимать под словами "вычислимая функция". Если понимать

их в расплывчато-интуитивном смысле ("функция вычисляется алгоритмически, то есть по чётким, недвусмысленным, однозначным правилам" или что-то в таком роде), конечно, ни о каком доказательстве тезиса Тьюринга не может быть речи. Можно лишь говорить, что многовековая практика человечества от Евклида до Кнута (ныне живущий крупнейший специалист в теории алгоритмических языков) не встретилась с примером алгоритма, который нельзя было бы записать как программу машины Тьюринга и т. п. Впрочем, ещё один (не слишком убедительный) аргумент приведён ниже.

Но если понимать слово "вычислимая" в тезисе Тьюринга как "вычислимая с помощью программы на C" и представить себе на минуту, что синтаксис и семантика C-программ точно определены, то тезис Тьюринга станет уже чётким утверждением, которое может быть истинным или ложным, и которое можно доказывать. Конечно, такое доказательство по необходимости должно использовать формальное описание синтаксиса и семантики C, и потому никем не проводилось, но для более простых вычислительных моделей это действительно можно формально доказать. Впрочем, такого рода доказательства сродни доказательству корректности длинной программы, и потому желающих их писать и тем более читать немного.

В заключении приведём обещанный аргумент в пользу того, что любая вычислимая функция вычислима на машине Тьюринга. Пусть есть функция, которую человек умеет вычислять. При этом, он, естественно, должен использовать карандаш и бумагу, так как количество информации, которое он может хранить "в уме", ограничено. Будем считать, что он пишет на отдельных листах бумаги. Помимо текущего листа, есть стопка бумаг справа и стопка слева; в любую из них можно положить текущий лист, завершив с ним работу, а из другой стопки взять следующий. У человека есть карандаш и ластик. Поскольку очень мелкие буквы на листе неразличимы, число отчётливо различных состояний листа конечно, так что можно считать, что в каждый момент на листе записана одна буква из некоторого конечного (хотя и весьма большого) алфавита. Человек тоже имеет конечную память, так что его состояние есть элемент некоторого конечного множества. При этом можно составить некоторую таблицу, в которой записано, чем кончится его работа над листом с данным содержимым, начатая в данном состоянии (что будет на листе, в каком состоянии будет человек и из какой пачки будет взят следующий лист). Теперь уже видно, что действия человека как раз соответствуют работе машины Тьюринга с большим (но конечным) алфавитом и большим (но конечным) числом внутренних состояний.

ГЛАВА XII

Арифметичность вычислимых функций

§1. Программы с конечным числом переменных

Мы хотим показать, что график всякой вычислимой функции является арифметическим множеством, то есть выразим формулой арифметики. Для этого удобно перейти от машин Тьюринга к другой модели, которую можно условно назвать машинами с конечным числом регистров.

Программа для такой машины использует конечное число переменных, значениями которых являются натуральные числа. Числа эти могут быть произвольного размера, так что машина реально имеет память неограниченного объёма. Программа состоит из нумерованных по порядку команд. Каждая команда имеет один из следующих видов:

- `a=0;`
- `a=b;`
- `a=b+1;`
- `a=b-1;`
- `goto met;`
- `if (a==0) goto met1; else goto met2;`
- `exit(0);`

Поскольку мы считаем, что значения переменных — натуральные (целые неотрицательные) числа, условимся считать разность $0 - 1$ равной 0 (впрочем, это не так важно — можно было бы считать это аварией).

Дойдя до команды `exit`, программа заканчивает работу.

Как и для машин Тьюринга, полезна некоторая практика программирования. Для тренировки напишем программу сложения двух чисел. Она помещает в `c` сумму чисел, которые были в переменных `a` и `b`. Такая программа на C имела бы вид

```
c=a;
/* инвариант: ответ = сумма текущих значений c и b */
while (b!=0)
{
   c++;
   b--;
}
```

Имитируя цикл с помощью операторов перехода, получаем программу для нашей машины:

```
1: c=a;
2: if (b==0) goto 6; else goto 3;
3: c++;
4: b--;
5: goto 2;
6: exit(0);
```

Теперь легко понять, как написать программы для вычитания, умножения (которое реализуется как цикл с повторным сложением), деления с остатком (как учил ещё Энгельс в забытой ныне книге "Диалектика природы", деление есть сокращённое вычитание), возведения в степень, проверки простоты, отыскания n-го простого числа и т. п. Вообще по сравнению с машинами Тьюринга этот язык более привычен и потому легче поверить, что на нём можно запрограммировать все алгоритмы.

Единственное, чего в нём реально не хватает — это массивов. Но это легко обойти, поскольку есть числа произвольного размера и нас не интересует число операций (как это принято в общей теории алгоритмов). Вместо массива битов мы можем хранить число, двоичной записью которого он является, а для массивов чисел воспользоваться, скажем, основной теоремой арифметики и хранить последовательность $\langle a, b, c, d, e\rangle$ как число $2^a3^b5^c7^d11^e$. При этом операции `a[i]=b` и `b=a[i]` заменяются на небольшие программы, которые содержат переменные `a`, `b`, `i` и ещё несколько переменных. (Частью этих программ является нахождение простого числа с заданным порядковым номером.)

Легко определить понятие вычислимой (в этой модели) функции. Пусть есть программа с двумя переменными `x` и `y` (и, ".',.&-., другими). Поместим в `x` некоторое число n, а в остальные переменные поместим нули. Запустим программу. Если она не остановится, то вычисляемая ей функция в точке n не определена. Если остановится, то содержимое переменной `y` после остановки и будет значением функции, вычисляемой нашей программой (в точке n). Функция называется вычислимой (в этой модели), если существует вычисляющая её программа.

Как всегда, при определении мы фиксировали различные детали, большинство из которых не являются существенными. Можно было бы добавить некоторые команды (сложение, например) — или даже исключить (например, без копирования можно обойтись с помощью небольшой хитрости).

Задача 190. *Покажите, что класс вычислимых функций не изменится, если исключить из определения команду копирования* `a=b`.

Несколько более удивительно, что число переменных можно ограничить, скажем, сотней — но и это не так уж странно, если вспомнить, что мы можем хранить в одной переменной целый массив.

Задача 191. *Проверьте это.*

§2. Машины Тьюринга и программы

Построенная вычислительная модель не слабее машин Тьюринга в том смысле, что любую вычислимую на машинах Тьюринга функцию можно вычислить и программой с конечным числом переменных.

Теорема 68. *Всякая функция, вычислимая на машинах Тьюринга, может быть вычислена с помощью программы описанного вида с конечным числом переменных.*

Следует уточнить, однако, что мы имеем в виду, так как для машин Тьюринга исходное данное и результат были двоичными словами, а для программ — натуральными числами. Мы отождествляем те и другие по естественному правилу, при котором слова Λ (пустое), 0, 1, 00, 01,... соответствуют числам 0, 1, 2, 3, 4... (чтобы получить из числа слово, прибавим к нему единицу, переведём в двоичную систему и отбросим единицу в старшем разряде).

Доказательство. Как и раньше, мы приведём лишь приблизительное описание того, как по машине Тьюринга строится программа с конечным числом переменных, вычисляющая ту же функцию. Прежде всего конфигурации машины Тьюринга надо закодировать числами. Можно сделать это, например, поставив в соответствие каждой конфигурации четыре числа: номер текущего состояния, номер текущего символа (в ячейке, где стоит головка машины), код содержимого ленты слева от головки и код содержимого ленты справа от головки.

Чтобы решить, как удобнее кодировать содержимое ленты слева и справа от головки, заметим, что машина Тьюринга обращается с двумя половинами лентами слева и справа от головки как со стеками. (Стеком называется структура данных, напоминающая стопку листов. В неё можно положить лист наверх, взять верхний лист, а также проверить, есть ли ещё листы.) В самом деле, при движении головки направо из правого стека берётся верхний элемент, а в левый стек кладётся; при движении налево — наоборот. Стеки легко моделировать с помощью чисел: например, если в стеке хранятся символы 0 и 1, то добавление нуля соответствует операции $x \mapsto 2x$, добавление единицы — операции $x \mapsto 2x+1$, верхний элемент есть остаток

при делении на 2, а удаление верхнего элемента есть деление на 2 (с отбрасыванием остатка). Другими словами, мы воспринимаем двоичную запись числа как стек, вершина которого находится справа, у младшего разряда. Точно так же можно использовать n-ичную систему счисления и представить стек с n возможными символами в каждой позиции.

Теперь основной цикл машины Тьюринга можно записать как программу, оперирующую с указанными четырьмя числами (символ, состояние, левый стек и правый стек) — без особых хитростей. Но несколько вещей всё-таки надо иметь в виду.

Во-первых, стеки конечны, а лента бесконечна — мы должны договориться, что если стек опустошается, то в него автоматически добавляется символ пробела. Тем самым бесконечный пустой хвост ленты может присутствовать в стеке, как теперь говорят, виртуально.

Во-вторых, напомним, что мы договорились отождествлять двоичные слова (которые подаются на вход машины Тьюринга) и их коды (которые хранятся в переменных нашей программы). Поэтому, получив код входного слова, надо его разобрать по символам и положить эти символы один за другим в стек (основания систем счисления разные, так что просто так переписать это нельзя). Аналогичные проблемы возникают и при превращении выхода (части содержимого правого стека) в соответствующее число, но все они легко преодолимы, и мы не будем вдаваться в подробности. □

Верно и обратное утверждение:

Теорема 69. *Всякая функция, вычислимая программой с конечным числом переменных, вычислима на машине Тьюринга.*

Доказательство. Нам надо моделировать поведение программы с помощью машины Тьюринга. Будем считать, что значения переменных записаны на ленте (в двоичной системе) и разделены специальным разделительным символом. Тогда машина может найти любую переменную, идя от начала ленты и считая разделительные символы, сделать что-то с этой переменной и затем вернуться обратно в начало. (Нет необходимости записывать на ленте номер исполняемой команды, поскольку команд конечное число и машина может помнить номер текущей команды как часть своего состояния.) Операции прибавления и вычитания единицы также легко выполнимы в двоичной записи (если идти справа налево). Надо только иметь в виду, что размер числа может увеличиться, и тогда нужно для него освободить место, сдвинув все символы справа от головки на одну позицию. (При уменьшении нужно сдвинуть влево.) Ясно, что это также легко выполнить с помощью машины Тьюринга.

Если мы записываем числа в двоичной системе, то проблемы с перекодированием при вводе-выводе минимальны (надо лишь дописать нули для значений остальных переменных в начале работы и встать в нужное место ленты в конце). □

§3. Арифметичность вычислимых функций

Сейчас мы докажем, что функции, вычислимые программами с конечным числом переменных, арифметичны, то есть их графики являются арифметическими множествами. В этом разделе мы вновь предполагаем некоторое знакомство читателя с логическими обозначениями, и будем рассматривать *арифметические формулы*, содержащие переменные по натуральным числам, равенство, константы 0 и 1, операции сложения и умножения, логические связки (И, ИЛИ, НЕ) и кванторы "для всех" и "существует". Формально говоря, мы рассматриваем сигнатуру, содержащую единственный двуместный предикатный символ (равенство), две константы 0 и 1 и два двуместных функциональных символа (сложение и умножение). Говоря об истинности таких формул, мы имеем в виду их истинность в стандартной интерпретации, носителем которой является множество $\mathbb{N}$ натуральных чисел.

Множество $A \subset \mathbb{N}^k$ называется *арифметическим*, если существует арифметическая формула α с параметрами $x_1, \ldots, x_k$, которая его представляет в следующем смысле: $\langle n_1, \ldots, n_k \rangle \in A$ тогда и только тогда, когда формула α истинна при значениях параметров $x_1 = n_1, \ldots, x_k = n_k$.

Теорема 70. *График любой функции, вычисляемой программой с конечным числом переменных, является арифметическим множеством.*

Доказательство. Пусть $f\colon \mathbb{N} \to \mathbb{N}$ — функция, вычислимая некоторой программой P с конечным числом переменных $k_1, \ldots, k_N$. Будем считать, что входной переменной является k_1, а выходной — k_2. Нам нужно написать формулу с двумя переменными x, y, которая была бы истинна тогда и только тогда, когда $y = f(x)$. Состояние программы с конечным числом переменных полностью описывается значениями переменных и номером текущей команды (в процессорах соответствующий регистр часто называют program counter, по-русски счётчик команд). Легко понять, что соответствие между двумя последовательными состояниями программы с конечным числом переменных арифметично. Мы имеем в виду, что можно написать арифметическую формулу

$$\mathrm{Step}(s_1, \ldots, s_N, p, s'_1, \ldots, s'_N, p'),$$

с $2N+2$ переменными, которая утверждает, что данная программа P из состояния, где переменные равны $s_1, \ldots, s_N$, а счётчик команд равен p, за один шаг переходит в состояние, где переменные равны $s'_1, \ldots, s'_n$, а счётчик команд равен p'. (Договоримся, что значение $p' = 0$ соответствует остановке программы.) Такая формула является конъюнкцией отдельных утверждений, соответствующих каждой строке программы. Пусть, например, строка 7 программы имеет вид $\mathtt{k_2{=}k_3}$. Тогда в конъюнкции будет член вида

$$(p = 7) \Rightarrow ((s'_1 = s_1) \wedge (s'_2 = s_3) \wedge (s'_3 = s_3) \wedge \ldots \wedge (s'_N = s_N) \wedge (p' = 8)).$$

Для строки с условными переходами типа

```
3:  if (k5==0) goto 17; else goto 33;
```

в формуле будет два конъюнктивных члена (на два случая перехода)

$$((p = 3) \wedge (s_5 = 0)) \Rightarrow ((s'_1 = s_1) \wedge \ldots \wedge (s'_N = s_N) \wedge (p' = 17))$$

и

$$((p = 3) \wedge (s_5 \neq 0)) \Rightarrow ((s'_1 = s_1) \wedge \ldots \wedge (s'_N = s_N) \wedge (p' = 33)).$$

Надо ещё добавить утверждение о том, что при $p = 0$ работа прекращается, то есть что переменные на следующем шаге сохраняют свои значения, и p' остаётся равным 0.

Таким образом, арифметичность одного шага работы программы доказать несложно. Остаётся главный вопрос: как записать в виде формулы тот факт, что существует *последовательность* шагов, которая начинается с исходного состояния, заканчивается в данном и в которой каждый шаг правилен. Трудность в том, что здесь нужно как бы написать переменное число кванторов существования — или квантор "существует конечная последовательность натуральных чисел".

Это делается с помощью приёма, традиционно называемого β-функцией Гёделя. Вот что имеется в виду.

Лемма 1. Для любого k можно найти сколь угодно большое целое положительное число b, при котором первые k членов последовательности $b+1, 2b+1, 3b+1, \ldots$ попарно взаимно просты.

Доказательство. Любой общий простой делитель двух из этих чисел будет делителем их разности, то есть числа lb при $0 < l < k$; взяв b кратным $k!$, мы гарантируем, что он будет делителем числа b, но все члены нашей последовательности взаимно просты с b. Лемма доказана.

Лемма 2. Для любой последовательности $x_0, x_1, \ldots, x_n$ натуральных чисел можно найти такие числа a и b, что x_i есть остаток от деления a на $b(i+1)+1$.

Доказательство. Согласно предыдущей лемме, делители $b(i+1)+1$ можно взять взаимно простыми (и сколь угодно большими), остаётся воспользоваться "китайской теоремой об остатках". Эта теорема утверждает, что если целые положительные числа $d_1, \ldots, d_k$ взаимно просты, то при делении целого u на них может получиться любой заданный набор остатков. В самом деле, таких наборов будет $d_1 d_2 \ldots d_k$ (поскольку при делении на d_i возможны остатки от 0 до $d_i - 1$). При делении чисел $u = 0, 1, \ldots, d_1 d_2 \ldots d_k - 1$ получаются разные наборы остатков (если два числа u' и u'' дают одинаковые остатки, то их разность делится на все d_i, что невозможно в силу взаимной простоты). Поэтому чисел столько же, сколько наборов остатков, и должны появиться все наборы. Лемма 2 доказана.

Эта лемма показывает, что последовательность произвольной длины можно закодировать тремя числами a, b и n (последнее число — длина последовательности). Таким образом, условно говоря, можно заменить "формулу"

$$\exists \langle x_0, \ldots, x_n \rangle (\forall i \leqslant n)[\ldots x_i \ldots]$$

(которая на самом деле не является арифметической формулой, так как содержит квантор по конечным последовательностям) на формулу

$$\exists a\, \exists b\, \exists n\, (\forall i \leqslant n)[\ldots (\text{остаток от деления } a \text{ на } b(i+1)+1) \ldots].$$

Мы будем записывать остаток от деления a на $b(i+1)+1$ как $\beta(a, b, i)$ (отсюда и название "бета-функция").

Возвращаясь к программе P с конечным числом переменных $k_1, \ldots, k_N$ и вычисляемой ей функции f, можно записать утверждение вида $f(x) = y$ так: существуют такое число шагов n и такие числа $a_1, b_1, a_2, b_2, \ldots, a_N, b_N, a, b$, что

- $\beta(a_1, b_1, 0), \ldots, \beta(a_N, b_N, 0)$ есть правильные начальные значения переменных (первое равно x, остальные равны 0); $\beta(a, b, 0)$ есть правильное начальное значения счётчика команд, то есть 1;
- для каждого i от 0 до $n-1$ имеет место

$$\mathrm{Step}(\beta(a_1, b_1, i), \ldots, \beta(a_N, b_N, i), \beta(a, b, i), \\ \beta(a_1, b_1, i+1), \ldots, \beta(a_N, b_N, i+1), \beta(a, b, i+1)),$$

  то есть каждый переход соответствует программе;
- $\beta(a_2, b_2, n) = y$ (значение выходной переменной k_2 в конце вычисления равно y) и $\beta(a, b, n) = 0$ (значение счётчика команд в конце вычисления равно 0, что по нашей договорённости соответствует остановке машины).

Итак, арифметичность вычислимых (на машинах с конечным числом переменных) функций доказана. □

Вспоминая теорему 68, мы заключаем, что всякая вычислимая на машине Тьюринга функция арифметична. Принимая тезис Тьюринга, можно сказать, что график любой вычислимой функции является арифметическим множеством.

§4. Теоремы Тарского и Гёделя

Поскольку графики вычислимых функций арифметичны, очевидно, разрешимые и перечислимые множества тоже будут арифметическими.

Рассмотрим теперь множество T, элементами которого являются все истинные арифметические формулы без параметров (точнее, их номера в какой-то вычислимой нумерации всех формул.)

Теорема 71. *Множество T не арифметично.*

Доказательство. Начиная доказывать теорему Тарского, предположим, что множество номеров всех истинных арифметических формул (без параметров) арифметично. Пусть T — это множество, а $\tau(x)$ — соответствующая формула. Перенумеруем также все формулы с одним параметром x; пусть $F_n(x)$ — формула, имеющая номер n в этой нумерации. Рассмотрим формулу с единственным параметром x, утверждающую, что результат подстановки константы x в x-ую формулу с параметром ложен. Эту формулу можно написать так:

$$\exists z(\neg\tau(z) \wedge \mathrm{Subst}(z, x, x)),$$

где $\mathrm{Subst}(p, q, r)$ — формула с тремя параметрами, выражающая такое свойство: "p есть номер (в нумерации всех формул без параметров) той формулы, которая получится, если в q-ю формулу с одним параметром подставить константу r вместо этого параметра". Записанное в кавычках свойство описывает график некоторой вычислимой функции (соответствующей простым синтаксическим действиям и переходу от одной нумерации к другой) и потому существует выражающая его формула.

Итак, мы написали некоторую формулу с единственным параметром x. Пусть она имеет некоторый номер N. Подставим этот номер N вместо её параметра. Получится некоторая формула без параметров. Из построения видно, что эта формула истинна тогда и только тогда, когда результат подстановки числа N в формулу номер N (то есть сама эта формула!) ложен.

Это противоречие завершает доказательство теоремы Тарского. Мы видим, что нам потребовалась выразимость в арифметике не любых вычислимых функций, а одной вполне конкретной. При достаточном терпении соответствующую формулу можно-таки написать, и тем самым доказательство

станет совсем "осязаемым".

□

Эта теорема называется *теоремой Тарского*. Её можно прочесть так: множество арифметических истин не арифметично. Или: понятие арифметической истины невыразимо в арифметике.

Задача 192. *Покажите, что для любого N множество всех истинных замкнутых арифметических формул, содержащих не более N кванторов, арифметично.*

Задача 193. *Сформулируйте и докажите аналогичное утверждения для формул ограниченной кванторной глубины (число вложенных кванторов) и для формул с ограниченным числом перемен кванторов в предварённой нормальной форме.*

Теорема 72. *Множество T арифметических истин неперечислимо.*

Доказательство. В самом деле, любое перечислимое множество арифметично. □

Это утверждение называется *теоремой Гёделя о неполноте*. Его можно переформулировать так: всякое исчисление, порождающее формулы арифметики (т. е. алгоритм, перечисляющий некоторое множество таких формул) либо *неадекватно* (порождает некоторую ложную формулу), либо *неполно* (не порождает некоторой истинной формулы).

Теперь изложим доказательство теоремы Гёделя. Как мы уже говорили, исчисление — это механизм (алгоритм), который позволяет порождать некоторые формулы языка арифметики (для простоты будем считать, что порождаются только формулы без параметров). Таким образом, возникает некоторое перечислимое множество, которое обычно задают как проекцию разрешимого множества. Именно, вводят некоторое понятие *доказательства*. При этом доказательства являются словами в некотором алфавите. Множество доказательств разрешимо, то есть есть алгоритм, отличающий настоящие доказательства от текстов, который таковыми не являются. Кроме того, есть (также разрешимое) свойство двух слов x и y, которое гласит, что x есть доказательство формулы y. Перенумеровав все доказательства и формулы и выразив указанные разрешимые свойства в языке арифметики, мы приходим к формуле $\mathrm{Proof}(x, y)$, которая истинна, когда x есть номер доказательства формулы с номером y.

Теперь напишем формулу с одним параметром x, которая говорит, что результат подстановки числа x вместо параметра в x-ую формулу с одним

параметром не имеет доказательства:

$$\neg\exists z\exists p[\mathrm{Subst}(z,x,x)\wedge\mathrm{Proof}(p,z)]$$

Эта формула имеет единственный параметр (x); пусть её номер в нумерации таких формул равен N. Подставим N вместо параметра. Получится формула без параметров φ. По построению формула φ истинна, когда результат подстановки N в N-ю формулу с одним параметром недоказуем. А этот результат есть сама формула φ, так что она истинна тогда и только тогда, когда недоказуема. Значит, наше исчисление либо позволяет доказать ложную формулу φ (если φ ложна; в таком случае его называют неадекватным), либо не позволяет доказать истинную формулу φ.

Заметим, что доказательства Теорем Геделя и Тарского напоминают как построение неподвижной точки, так и классический парадокс лжеца:

УТВЕРЖДЕНИЕ В РАМКЕ ЛОЖНО

§5. О непостижимой эффективности математики

Изложенные в этой главе результаты, при внешней сухости, обычно свойственной математическим текстам, в свое время вызвали потрясение основ всей математики, да и естествознания в целом.

Наибольшее смятение у математиков вызвал результат Гёделя, утверждающий, что непротиворечивость любой достаточно мощной математической системы, охватывающей арифметику целых чисел, не может быть установлена средствами самой этой системы на основе математических принципов, принятых различными школами в основаниях математики. Результаты Гёделя послужили поводом для известного высказывания Германа Вейля: *"Бог существует, поскольку математика, несомненно, непротиворечива, но существует и дьявол, поскольку доказать ее непротиворечивость мы не можем."*

Приведенный результат Гёделя является следствием из теоремы Гёделя о неполноте. Она утверждает, как мы видели выше, что если формальная теория , включающая арифметику целых чисел, непротиворечива, то она неполна. Иначе говоря, существует имеющее смысл утверждение арифметики целых чисел (обозначим его S), которое в рамках данной теории невозможно ни доказать, ни опровергнуть. Но либо утверждение S, либо утверждение *не* S истинно. Следовательно, в арифметике существует истинное утверждение, которое недоказуемо, а значит, и неразрешимо. Хотя Гёдель не указал точно, о каком классе аксиоматических систем идет речь в полученном им результате, теорема о неполноте применима к системам

Рассела - Уайтхеда, Цермело - Френкеля, гильбертовской аксиоматике чисел и ко всем наиболее распространенным аксиоматическим системам.

Казалось, непротиворечивость достигается ценой неполноты Осуществив перевод словесных утверждений метаматематики на арифметический язык, Гёдель показал, как построить арифметическое утверждение G, означающее в переводе на метаматематический язык, что утверждение с гёделевским номером g недоказуемо. Но утверждение G, рассматриваемое как последовательность символов, имеет гёделевский номер g. Следовательно, G утверждает о самом себе, что оно недоказуемо. Итак, если G доказуемо, то оно должно быть недоказуемым, а если G недоказуемо, то оно должно быть доказуемым, поскольку недоказуемо, что оно недоказуемо. Так как любое арифметическое утверждение либо истинно, либо ложно, формальная система, которой принадлежит G, неполна (если только она непротиворечива). Тем не менее арифметическое утверждение G истинно, так как является утверждением о целых числах, которое можно доказать, используя более интуитивные рассуждения, чем допускает формальная система. Поясним суть гёделевской схемы на примере. Рассмотрим, утверждение S: "Это утверждение ложно" (парадокс лжеца, другая эквивалентная форма). Оно приводит к противоречию. Гёдель заменил слово *ложно* словом *недоказуемо*, превратив S в утверждение G - *Это утверждение недоказуемо*. Если утверждение недоказуемо, то утверждаемое им истинно. С другой стороны, если утверждение доказуемо, то оно ложно, или, в соответствии с обычной логикой, если утверждение истинно, то оно недоказуемо. Следовательно, утверждение истинно в том и только в том случае, если оно недоказуемо. Мы приходим не к противоречию, а к истинному утверждению, которое недоказуемо, т. е. неразрешимо.

Заготовив впрок неразрешимое утверждение, Гёдель построил арифметическое утверждение ψ, соответствующее метаматематическому утверждению *Арифметика непротиворечива*, и доказал, что из ψ следует G. Поэтому если бы ψ было доказуемым, то и G было бы доказуемым. Но так как G неразрешимо, ψ недоказуемо. Иными словами, утверждение A неразрешимо. Тем самым установлена невозможность доказать *внутренними средствами* (т. е. в рамках той же системы) непротиворечивость арифметики любым методом — с помощью любой системы логических принципов, представимой в виде арифметической системы.

На первый взгляд кажется, что неполноты можно было бы избежать, если ввести в формальную систему дополнительный логический принцип или математическую аксиому. Но метод Гёделя позволяет доказать, что если дополнительное утверждение допускает перевод на язык арифметики по предложенной Гёделем схеме (согласно которой символам и формулам мы

ставим в соответствие некоторые числа - их гёделевские номера), то и в расширенной системе можно сформулировать неразрешимое утверждение. Иначе говоря, избежать неразрешимых утверждений и доказать непротиворечивость можно лишь с помощью логических принципов, *не отображаемых* в арифметику. Чтобы пояснить суть дела, воспользуемся аналогией (хотя и несколько неточной): если бы логические принципы и математические аксиомы были сформулированы на японском языке, а арифметизация Гёделя означала бы перевод на английский язык, то результаты Гёделя получались бы до тех пор, пока был бы осуществим перевод с японского на английский.

Таким образом, теорема Гёделя о неполноте утверждает, что ни одна система математических и логических аксиом, арифметизуемая тем или иным способом (например, так, как это сделал Гёдель), не позволяет охватить даже все содержащиеся в ней истины, не говоря уже о всей математике, поскольку любая система аксиом неполна. В любой аксиоматической системе существуют утверждения, недоказуемые в рамках данной системы. Истинность таких утверждений может быть установлена лишь с помощью неформальных рассуждений.

Теорема Гёделя о неполноте, показавшая, что аксиоматизация имеет свои пределы, разительно отличалась от господствовавших в конце XIX в. представлений о математике как о совокупности аксиоматизируемых (и аксиоматизированных) теорий. Теорема Гёделя нанесла сокрушительный удар по всеобъемлющей аксиоматизации. Неадекватность аксиоматического подхода сама по себе противоречием не была; однако она явилась полной неожиданностью, поскольку математики, особенно формалисты, предполагали, что в рамках некоторой аксиоматической системы любое истинное в ней утверждение заведомо доказуемо. Браузр установил, что интуитивно воспринимаемые истины часто лежат далеко за пределами того, что было доказано в классической математике, а Гёдель доказал, что интуитивно воспринимаемые истины вообще выходят за рамки математического доказательства. По выражению Пауля Бернайса, ныне более разумно не столько рекомендовать аксиоматику, сколько предостерегать против ее переоценки, разумеется, сказанное выше не исключает возможности появления новых методов доказательства, которые выходят за пределы допустимого логическими принципами, принятыми различными школами в основаниях математики.

Оба полученных Гёделем результата потрясли математику. Это означало, что математика вынуждена бесповоротно отказаться от претензий на абсолютную достоверность или значимость своих результатов, т. е. лишиться одной из основных своих особенностей, на которую претендовала

еще сравнительно недавно.

Положение осложнялось невозможностью доказать непротиворечивость: ведь все, о чем говорили математики, могло оказаться бессмыслицей, ибо теперь никто не мог гарантировать, что в будущем не возникнет противоречия. Случись такое и окажись противоречие неразрешимым - вся математика обратилась бы в прах. Действительно, одно из двух противоречивых утверждений должно быть ложным, а согласно принятой всеми математическими логиками концепции импликации, из ложного утверждения может следовать что угодно. Итак, математики работали под угрозой полного провала. Еще один удар нанесла теорема о неполноте.

Теорему Гёделя о неполноте до некоторой степени можно рассматривать как отрицание закона исключенного третьего. Каждое утверждение мы считаем либо истинным, либо ложным. В современных основаниях математики это означает, что рассматриваемое утверждение доказуемо или недоказуемо с помощью законов логики и аксиом того раздела математики, к которому относится интересующее нас утверждение. Гёдель же доказал, что некоторые утверждения нельзя ни доказать, ни опровергнуть. Непротиворечивость можно было бы считать доказанной, если бы в противовес подходу Гёделя в системе удалось обнаружить неразрешимое утверждение: ведь как мы уже установили, опираясь на свойства материальной импликации, если в системе имеется противоречие, то в ней можно доказать что угодно. Однако до сих пор обнаружить неразрешимое утверждение не удалось.

Лауреат Нобелевской премии по физике Юджин Пол Вигнер, обсуждая в 1960 г. непостижимую эффективность математики в естественных науках в статье под тем же названием, не дал ника кого объяснения и ограничился лишь констатацией спорного вопроса: *"Математический язык удивительно хорошо приспособлен для формулировки физических законов. Это чудесный дар, который мы не понимаем и которого не заслуживаем. Нам остается лишь благодарить за него судьбу и надеяться, что и в будущих своих исследованиях мы сможем по-прежнему пользоваться им. Мы думаем, что сфера его применимости (хорошо это или плохо) будет непрерывно возрастать, принося нам не только радость, но и новые головоломные проблемы."*

Замечательная точность и эффективность математики в описании реального мира по-прежнему ждут своего объяснения. Несмотря на обнаруженную ограниченность возможностей, математике есть чем гордиться. Она была и остается высшим интеллектуальным достижением и наиболее оригинальным творением человеческого духа. Музыка может возвышать или умиротворять душу, живопись - радовать глаз, поэзия — пробуждать чув-

ства, философия - удовлетворять потребности разума, инженерное дело - совершенствовать материальную сторону жизни людей. Но математика способна достичь всех этих целей. Если же говорить о возможностях человеческого разума, то математики немало потрудились, чтобы доказать, сколь высокую надежность результатов способен обеспечить человеческий разум. Не случайно математическая точность вошла в поговорку. Математика попрежнему остается эталоном самого надежного и точного знания, которого мы только в состоянии достичь.

Все свершения математики - это свершения человеческого разума. Показав, на что способен человек, математика вселила в людей уверенность, позволившую им вплотную взяться за разгадку ранее, казалось бы, неприступных тайн космоса, лечение страшных болезней, количественный анализ проблем, относящихся к экономике и устройству человеческого общества, что позволяет надеяться на дальнейший прогресс человечества. В решении этих проблем именно с математикой связаны основные надежды на успех.

ГЛАВА XIII

Рекурсивные функции

§1. Примитивно рекурсивные функции

Программы с конечным числом переменных напоминали ассемблер; рассматриваемые в этом разделе рекурсивные функции скорее напоминают функциональное программирование, когда одни функции определяются через другие. Мы будем рассматривать функции с натуральными аргументами и значениями. Вообще говоря, функции могут быть не всюду определенными, так что говоря о функции n аргументов (функции из $\mathbb{N}^n$ в $\mathbb{N}$, n-местной функции), мы имеем в виду функцию, определённую на некотором подмножестве $\mathbb{N}^n$ со значениями в $\mathbb{N}$.

Пусть имеется одна k-местная функция f и k штук n-местных $g_1, \ldots, g_k$. Тогда из них можно сформировать одну n-местную функцию

$$\langle x_1, \ldots, x_n \rangle \mapsto f(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)).$$

Говорят, что определённая таким образом функция получена из функций f и $g_1, \ldots, g_k$

с помощью операции *подстановки*.

Другая операция, называемая операцией *рекурсии*, или

примитивной рекурсии, применяется к k-местной функции f и $(k+2)$-местной функции g. Её результатом будет $(k+1)$-местная функция h, определяемая так:

$$h(x_1, \ldots, x_k, 0) = f(x_1, \ldots, x_k); \tag{1}$$

$$h(x_1, \ldots, x_k, y+1) = g(x_1, \ldots, x_k, y, h(x_1, \ldots, x_k, y)). \tag{2}$$

В последовательности $h(x_1, \ldots, x_n, 0), h(x_1, \ldots, x_n, 1), \ldots$ каждое значение определяется через предыдущее, поэтому если какое-то из значений не определено, то не определены и все последующие.

Для единообразия будем считать, что нуль-местные функции (функции без аргументов) суть константы; это позволяет рекурсивно определять функции одной переменной.

Примитивно рекурсивными называют функции, которые можно получить с помощью операций подстановки и рекурсии из следующих *базисных* функций: константы 0, операции прибавления единицы $s\colon x \mapsto x+1$ и семейства функций проекции: это семейство для каждого k содержит k штук k-местных функций $\pi^i_k(x_1, \ldots, x_k) = x_i$.

Функции проекции позволяют выполнять "неоднородные" подстановки: скажем, можно получить функцию $\langle x, y\rangle \mapsto f(g(x), h(y, x, y), x)$ из функций f и h, комбинируя их с функциями проекции: сначала получаем функцию $\langle x, y\rangle \mapsto g(x)$ (подстановка π^1_2 в g), затем $\langle x, y\rangle \mapsto h(y, x, y)$ (подстановка $\pi^2_2, \pi^1_2, \pi^2_2$ в h), затем полученные две функции вместе с функцией π^1_2 подставляем в f.

Подставляя константу 0 в функцию прибавления единицы, получаем константу (функцию нуля аргументов) 1. Затем можно получить константы 2, 3 и т. д.

§2. Примеры примитивно рекурсивных функций

Как и с другими вычислительными моделями, важно накопить некоторый программистский опыт.

Сложение. Функция $\langle x, y\rangle \mapsto \mathrm{sum}(x, y) = x + y$ получается с помощью рекурсии:

$$\mathrm{sum}(x, 0) = x; \tag{1}$$

$$\mathrm{sum}(x, y + 1) = \mathrm{sum}(x, y) + 1. \tag{2}$$

Надо, конечно, представить правую часть второго равенства как результат подстановки. Формально говоря, $h(x, y, z)$ в определении рекурсии надо положить равным $s(z)$, где s — функция прибавления единицы.

Умножение. Функция $\langle x, y\rangle \mapsto \mathrm{prod}(x, y) = xy$ получается с помощью рекурсии (с использованием сложения):

$$\mathrm{prod}(x, 0) = 0; \tag{3}$$

$$\mathrm{prod}(x, y + 1) = \mathrm{prod}(x, y) + x. \tag{4}$$

Аналогичным образом можно перейти от умножения к возведению в степень.

Усечённое вычитание. Мы говорим об "усечённом вычитании" $x \dot{-} y = x - y$ при $x \geqslant y$ и $x \dot{-} y = 0$ при $x < y$, поскольку мы имеем дело только с натуральными (целыми неотрицательными) числами. Одноместная функция усечённого вычитания единицы определяется рекурсивно:

$$0 \dot{-} 1 = 0; \tag{5}$$

$$(y + 1) \dot{-} 1 = y. \tag{6}$$

(Рекурсия здесь формальна, так как предыдущее значение не используется.) После этого усечённое вычитание для произвольных аргументов можно

определить так:

$$x \dot{-} 0 = x; \tag{7}$$

$$x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1. \tag{8}$$

§3. Примитивно рекурсивные множества

Будем называть множество *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна. (Вариант: если оно является множеством нулей примитивно рекурсивной функции; это то же самое, так как можно сделать подстановку в функцию $x \mapsto 1 \dot{-} x$.)

Пересечение и объединение примитивно рекурсивных множеств примитивно рекурсивны (сложим или перемножим функции, множествами нулей которых они являются). Дополнение примитивно рекурсивного множества примитивно рекурсивно. Отождествляя множества со свойствами, можно сказать, что конъюнкции, дизъюнкции и отрицания примитивно рекурсивных свойств будут примитивно рекурсивны.

Свойства $x = y$ и $x \neq y$ примитивно рекурсивны ($x = y$ тогда и только тогда, когда $(x \dot{-} y) + (y \dot{-} x) = 0$).

Функция $f(x)$, заданная соотношением

$$f(x) = [\ \textbf{if}\ (R(x))\ g(x);\ \textbf{else}\ h(x);\ ],$$

будет примитивно рекурсивной, если таковы функции g и h и свойство R. В самом деле, $f(x)$ можно записать как $r(x)g(x) + (1 \dot{-} r(x))h(x)$, где r — характеристическая функция свойства R.

Теперь можно записать формулу для прибавления единицы по модулю n (для чисел, меньших n):

$$x + 1 \bmod n = [\ \textbf{if}\ (x + 1 == n)\ 0;\ \textbf{else}\ x + 1;\ ]$$

После этого функцию $x \bmod n$ (остаток от деления на n) можно определить рекурсивно:

$$0 \bmod n = 0; \tag{1}$$

$$(x+1) \bmod n = (x \bmod n) + 1 \bmod n. \tag{2}$$

Покажем, что ограниченные кванторы, применённые к примитивно рекурсивным свойствам (множествам), дают снова примитивно рекурсивные свойства. Это означает, например, что если свойство $R(x, y)$ примитивно рекурсивно, то свойства

$$S(x, z) = (\exists y \leqslant z)\ R(x, y)$$

и

$$T(y,z) = (\forall y \leqslant z)\, R(x,y)$$

также примитивно рекурсивны. Чтобы убедиться в этом, заметим, что для функций ограниченный квантор соответствует перемножению или суммированию: если свойство $R(x,y)$ равносильно $r(x,y) = 0$, то

$$S(x,z) \Leftrightarrow \left[\prod_{y=0}^{z} r(x,y) = 0\right].$$

А произведение легко определить рекурсивно:

$$\prod_{y=0}^{0} r(x,y) = r(x,0); \tag{3}$$

$$\prod_{y=0}^{t+1} r(x,y) = \left[\prod_{y=0}^{t} r(x,y)\right] \cdot r(x,t+1); \tag{4}$$

с суммированием можно поступить аналогичным образом.

После этого легко заметить, что свойство "быть простым" примитивно рекурсивно (любое меньшее число либо равно нулю, либо равно 1, либо не является делителем).

Покажем теперь, что если график некоторой функции f примитивно рекурсивен и её значения ограничены сверху некоторой примитивно рекурсивной функцией g, то сама функция f примитивно рекурсивна. В самом деле, если r — характеристическая функция графика, то есть $r(x,y) = 1$ при $y = f(x)$ и $r(x,y) = 0$ при $y \neq f(x)$ (для простоты мы рассматриваем случай функций одного аргумента), то

$$f(x) = \sum_{i=0}^{\infty} y \cdot r(x,y),$$

а суммирование можно ограничить сверху выражением $g(x)$ и воспользоваться примитивной рекурсивностью ограниченной суммы.

Отсюда легко вывести следующее утверждение: если функция g и свойство $R(x,y)$ примитивно рекурсивны, то функция

$$x \mapsto f(x) = \text{наименьшее } y \leqslant g(x), \text{ для которого } R(x,y)$$

(если для некоторого x такого y нет, то полагаем значение функции равным, скажем, $g(x)+1$) будет примитивно рекурсивной. В самом деле, график функции f легко описать с помощью ограниченных кванторов.

Такой способ определения функции называют *ограниченным оператором минимизации* — в отличие от неограниченного, где нет заранее известной границы $g(x)$. Как мы увидим, в неограниченном случае получающаяся функция не обязана быть примитивно рекурсивной.

Ограниченный оператор минимизации можно использовать, чтобы убедиться, что функция $x \mapsto$ (минимальное простое число, большее x) примитивно рекурсивна (рассуждение Евклида о бесконечности множества простых чисел устанавливает, что это число не превосходит $x! + 1$, а факториал примитивно рекурсивен). После этого функция $n \mapsto$ (n-е простое число) легко определяется с помощью рекурсии.

§4. Другие виды рекурсии

Слова "рекурсивное определение функции" можно понимать и в более широком смысле, нежели мы это делали (см. выше определение рекурсии, или примитивной рекурсии) — как любой способ задания функции, который связывает значение функции в данной точке с другими её значениями. Как мы увидим ниже при обсуждении функции Аккермана, есть такие схемы рекурсивных определений, которые выводят из класса примитивно рекурсивных функций. Но есть и такие, которые можно свести к рассмотренной нами схеме.

Мы приведём два примера последнего типа: совместное определение нескольких функций и использование произвольных меньших значений аргумента.

Совместная рекурсия. Пусть две одноместные функции f и g заданы соотношениями:

$$f(0) = a, \tag{1}$$
$$g(0) = b, \tag{2}$$
$$f(n+1) = F(n, f(n), g(n)), \tag{3}$$
$$g(n+1) = G(n, f(n), g(n)), \tag{4}$$

где a и b — некоторые числа, а функции F и G — примитивно рекурсивные функции трёх аргументов. Покажем, что тогда функции f и g примитивно рекурсивны.

Чтобы доказать это, нам потребуется примитивно рекурсивная нумерация пар — такая функция $\langle x, y\rangle \to [x, y]$ (номер пары мы обозначаем квадратными скобками), которая была бы примитивно рекурсивна вместе с двумя обратными функциями (дающими по номеру пары её первый и второй члены). Тогда мы сможем написать рекурсивное определение для

функции $h(n) = [f(n), g(n)]$:

$$h(0) = [a, b], \tag{5}$$

$$h(n+1) = [F(n, p_1(h(n)), p_2(h(n))), \tag{6}$$

$$G(n, p_1(h(n)), p_2(h(n)))], \tag{7}$$

где функции p_1 и p_2 дают по номеру пары первый и второй её члены. Если функция h примитивно рекурсивна, то и функции f и g (композиции h с функциями p_1 и p_2) также примитивно рекурсивны.

Осталось объяснить, как найти примитивно рекурсивную нумерацию пар. Можно заметить, что есть многочлен второго порядка с двумя переменными, задающий взаимно однозначное соответствие $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Это соответствие видно из таблицы:

6			
3	7		
1	4	8	
0	2	5	9

Примитивную рекурсивность обратных отображений p_1 и p_2 можно установить, воспользовавшись ограниченной минимизацией, так как $p_1(n)$ есть минимальное $x \leqslant n$, для которого найдётся $y \leqslant n$, при котором $[x, y] = n$.

Менее симметричная нумерация пар может быть задана формулой $[a, b] = (2a + 1)2^b$. Можно также заметить, что нам не нужно, чтобы все числа были номерами каких-то пар, и воспользоваться нумерацией $[a, b] = 2^a 3^b$.

Заметим в заключение, что аналогичная конструкция применима для большего числа одновременно определяемых функций и для функций от большего числа аргументов.

Возвратная рекурсия. Следующее утверждение показывает, что при рекурсивном определении можно использовать не только значение в предыдущей точке, но и любое предшествующее значение.

Теорема 73. *Пусть функция g одного аргумента примитивно рекурсивна, причём $g(x) < x$ при $x > 0$; пусть F — примитивно рекурсивная функция двух аргументов; пусть c — произвольная константа. Тогда функция h, определённая соотношениями*

$$h(0) = c, \tag{8}$$

$$h(x) = F(x, h(g(x))) \text{ при } x > 0 \tag{9}$$

примитивно рекурсивна.

$\lhd$ Чтобы доказать эту теорему, используем следующую нумерацию конечных последовательностей натуральных чисел: номером пустой последо-

вательности считаем число 1, номером одноэлементной последовательности $\langle a\rangle$ считаем число 2^{a+1}, последовательность $\langle a,b\rangle$ имеет номер $2^{a+1}3^{b+1}$, последовательность $\langle a,b,c\rangle$ имеет номер $2^{a+1}3^{b+1}5^{c+1}$ и так далее (основания степеней — простые числа). Будем обозначать номер последовательности $\langle a,b,\ldots,z\rangle$ через $[a,b,\ldots,z]$. Эта нумерация в некотором смысле примитивно рекурсивна. Конечно, буквально это понимать нельзя, так как нумерация представляет собой "функцию с переменным числом аргументов". Но разные связанные с ней функции примитивно рекурсивны. В частности, таковы функции

- $\mathrm{Length}(x)$ = длина последовательности с номером x;
- $\mathrm{Select}(i,x)$ = i-ый член последовательности с номером x;
- $\mathrm{Append}(x,y)$ = номер последовательности, которая получается приписыванием числа y к последовательности с номером x.

Все эти функции (и другие аналогичные) сводятся к различным операциям с простыми числами и множителями, которые мы в сущности уже разбирали.

Теперь мы докажем, что функция

$$x \mapsto H(x) = [h(0), h(1), \ldots, h(x)]$$

примитивно рекурсивна. В самом деле, $H(0) = [c]$, а

$$H(k+1) = \mathrm{Append}(H(k), F(k+1, \mathrm{Select}(g(k+1), H(k)))). \triangleright$$

§5. Машины Тьюринга и примитивно рекурсивные функции

Мы рассмотрели несколько различных приёмов построения примитивно рекурсивных функций. Тем не менее остаётся не вполне ясным, насколько этот класс широк. Сейчас мы покажем, что он включает в себя все достаточно быстро вычислимые функции.

Теорема 74. *Любая функция, вычислимая на машине Тьюринга не более чем за примитивно рекурсивное (от длины входа) время, примитивно рекурсивна.*

Доказательство. Напомним, что мы считаем входом и выходом машины Тьюринга слова из нулей и единиц. Поскольку аргументами и значениями примитивно рекурсивных функций являются числа, теорема будет иметь смысл, только если мы договоримся отождествлять числа и слова. Как уже говорилось, мы отождествляем число n со словом, которое получается после удаления старшего бита 1 в двоичном разложении числа $n+1$.

При имитации работы машин Тьюринга с помощью программ мы кодировали состояние машины четырьмя числами (код левой части ленты, код правой части ленты, состояние и буква под головкой). При этом удобно было такое кодирование: левую часть ленты мы считали записью числа в системе счисления, в которой основание равно числу символов в алфавите машины, а пробел считается нулём; с правой частью ленты мы поступали так же, только в обратном порядке (младшие разряды у головки). При этом добавление или изъятие символа у головки соответствовало простой арифметической операции (удаление — это деление нацело, добавление — умножение на основание системы счисления и сложение). При таком кодировании функции перехода (четыре функции четырёх аргументов, показывающие следующее состояние как функцию предыдущего), записываются простыми формулами и примитивно рекурсивны.

Теперь рассмотрим итерированную функцию перехода, которая говорит, каково будет состояние машины Тьюринга после t шагов. Точнее, тут имеются четыре функции от пяти аргументов (первые четыре аргумента кодируют состояние, пятый представляет собой число шагов). Их определение имеет вид совместной рекурсии, которую мы только что разобрали. Поэтому эти функции примитивно рекурсивны. Будем считать, что после появления заключительного состоянии конфигурация машины не меняется. Если мы знаем, что число шагов работы ограничено примитивно рекурсивной функцией, то достаточно подставить её на место пятого аргумента (числа шагов), чтобы убедиться, что заключительная конфигурация машины является примитивно рекурсивной функцией от её начальной конфигурации. Следовательно, результат работы является примитивно рекурсивной функций начального данного.

Это рассуждение неявно использует примитивную рекурсивность различных функций, связанных с переходом от одного представления данных к другому. Например, вход машины Тьюринга является двоичным словом, которое мы договорились отождествлять с некоторым числом x. Этому входу соответствует начальная конфигурация машины Тьюринга, которую мы кодируем четвёркой чисел. Нам важно, что эта четвёрка примитивно рекурсивно зависит от x. Это легко понять, так как преобразование связано с переходом от одной системы счисления к другой (одно и то же слово кодирует разные числа в разных системах счисления); примитивную рекурсивность таких функций легко установить с помощью описанных выше методов. Кроме того, нам надо из выходной конфигурации примитивно рекурсивно извлечь результат и также перекодировать его, а также по входу получить его длину (чтобы подставить в примитивно рекурсивную функцию, ограничивающую число шагов). Но всё это также не выходит из круга

разобранных выше приёмов, и подробно останавливаться на этом мы не будем. □

Эта теорема убеждает нас в примитивной рекурсивности многих довольно сложно определяемых функций. Например, рассмотрим функцию $n \mapsto$ $\mapsto$ (n-ый десятичный знак числа π). Известно, что вычислены миллионы таких знаков, поэтому есть все основания полагать, что известные алгоритмы работают не слишком долго — было бы очень странно, если бы время их работы (даже учитывая неудобство машины Тьюринга для программирования) не оценивалось бы, скажем, функцией $c \times 2^n$ при достаточно большом c. А такая оценка примитивно рекурсивна, что позволяет сослаться на только что доказанную теорему. (На самом деле тут большой запас — существуют примитивно рекурсивные функции, которые растут гораздо быстрее 2^n.)

§6. Частично рекурсивные функции

Операторы примитивной рекурсии и подстановки не выводят нас из класса всюду определённых функций. Не так обстоит дело с *оператором минимизации*, о котором мы уже упоминали. Он применяется к $(k+1)$-местной функции f и даёт k-местную функцию g, определяемую так: $g(x_1, \ldots, x_k)$ есть *наименьшее y, для которого* $f(x_1, \ldots, x_k, y) = 0$.

Смысл выделенных слов ясен, если функция f всюду определена. Если нет, то понимать их надо так: значение $g(x_1, \ldots, x_k)$ равно y, если $f(x_1, \ldots, x_k, y)$ определено и равно нулю, а все значения $f(x_1, \ldots, y_k, y')$ при $y' < y$ определены и не равны нулю.

Часто используется обозначение

$$g(x_1, \ldots, x_k) = \mu y\ (f(x_1, \ldots, x_k, y) = 0),$$

и потому оператор минимизации также называют *μ-оператором*.

Ясно, что такое определение обеспечивает вычислимость g, если вычислима f (мы перебираем в порядке возрастания все y, ожидая появления нулевого значения).

Задача 194. *Покажите, что если изменить определение и разрешить $f(x_1, \ldots, x_k, y')$ быть не определённым при $y' < y$, то функция g может быть невычислимой при вычислимой f.*

Функции, получающиеся из базисных (нуля, проекции и прибавления единицы) с помощью операторов подстановки, примитивной рекурсии и минимизации, называются *частично рекурсивными*. Если такая функция оказывается всюду определённой, то её называют *общерекурсивной* функцией.

Теорема 75. *Всякая функция, вычислимая с помощью машины Тьюринга, является частично рекурсивной.*

Доказательство. Пусть f — вычислимая с помощью машины Тьюринга (обозначим эту машину через M) функция одного аргумента. Рассмотрим свойство $T(x, y, t)$, состоящее в том, что машина M на входе x даёт ответ y за время не более чем t. Как мы видели выше, по входу машины Тьюринга и по времени t можно примитивно рекурсивно вычислить её состояние в момент t; ясно, что можно также узнать, закончила ли она работу, и если да, то был ли ответ равен y. Итак, свойство T примитивно рекурсивно.

Теперь объединим аргументы y и t в пару с помощью примитивно рекурсивной нумерации; получится примитивно рекурсивная функция T', для которой $T'(x, [y, t]) = T(x, y, t)$; теперь можно написать $f(x) = p_1(\mu z T'(x, z))$, где p_1 даёт по номеру пары её первый член, а μz означает "наименьшее z, для которого ...". Таким образом, функция f является частично рекурсивной. □

Верно и обратное:

Теорема 76. *Всякая частично рекурсивная функция вычислима на машине Тьюринга.*

Доказательство. Легко написать программу с конечным числом переменных, вычисляющую любую частично рекурсивную функцию (подстановка сводится к последовательному выполнению программ, рекурсия — к циклу типа **for**, минимизация — к циклу типа **while**; оба вида циклов легко реализуются с помощью операторов перехода).

После этого остаётся только сослаться на то, что всякая функция, вычисляемая программой с конечным числом регистров, вычислима на машине Тьюринга (как мы видели в разделе 2, теорема 69). □

Поэтому если мы верим в "тезис Тьюринга", гласящий, что всякая вычислимая функция вычислима на машине Тьюринга, то должны верить и в "тезис Чёрча" (всякая вычислимая функция частично рекурсивна), так что эти тезисы равносильны.

Наше доказательство теорем 75 и 76 позволяет также получить такое следствие, называемое иногда *теоремой Клини о нормальной форме*:

Теорема 77. *Всякая частично рекурсивная функция f представима в виде*

$$f(x) = a(\mu z(b(x, z) = 0)),$$

где a и b — некоторые примитивно рекурсивные функции.

Доказательство. В самом деле, любая частично рекурсивная функция вычислима на машине Тьюринга, а следовательно, представима в нужном нам виде, как видно из доказательства теоремы 75 (в качестве a берётся функция, дающий первый член пары по её номеру). □

Мы сформулировали эту теорему для случая одноместной функции f, но аналогичное утверждение верно и для функций нескольких аргументов (и доказательство почти не меняется).

Задача 195. *Покажите, что одним μ-оператором, применяя его последним, не обойтись: не всякая частично рекурсивная функция представима в виде*

$$f(x) = \mu z(b(x, z) = 0)$$

где b — некоторая примитивно рекурсивная функция.

Из теоремы Клини о нормальной форме вытекает такое утверждение:

Теорема 78. *Всякое перечислимое множество есть проекция примитивно рекурсивного множества.*

Доказательство. Перечислимое множество есть область определения рекурсивной функции; представив её в нормальной форме, видим, что область определения есть проекция множества $\{\langle x, z\rangle \mid b(x, z) = 0\}$. □

§7. Оценки скорости роста. Функция Аккермана

Обратимся теперь к вопросу, который можно было бы задать уже давно: существуют ли общерекурсивные, но не примитивно рекурсивные функции? Мы приведём два доказательства существования таковых. Первое исходит из общих соображений:

Теорема 79. *Существует всюду определённая вычислимая функция двух аргументов, универсальная для класса всех примитивно рекурсивных функций одного аргумента.*

Очевидно, что если U — такая функция, то функция d, для которой $d(n) = U(n, n) + 1$, будет всюду определённой, вычислимой и будет отличаться от любой примитивно рекурсивной функции (от n-ой — в точке n).

Доказательство. Всякая примитивно рекурсивная функция получается из базисных с помощью некоторой последовательности операций подстановки и рекурсии. Ясно, что такую последовательность можно описать словом в конечном алфавите — так сказать, программой (в которой последовательно

определяются различные примитивно рекурсивные функции и для каждой написано, из каких других она получается и с помощью каких операций). Из всех программ отберём программы для одноместных функций (разумеется, в качестве промежуточных функций можно использовать функции с любым числом аргументов). Множество таких программ разрешимо, их можно пронумеровать вычислимым образом. Функция $\langle n, x\rangle \mapsto$ (результат применения функции, заданной программой номер n, к числу x) будет вычислима и по построению будет универсальной для класса примитивно рекурсивных функций. □

Однако интересно указать и более конкретную причину, мешающую некоторым вычислимым функциям быть примитивно рекурсивными. Вот одна из возможностей: примитивно рекурсивные функции не могут быстро расти. Эта идея восходит к Аккерману, который построил функцию, растущую быстрее всех примитивно рекурсивных — *функцию Аккермана*. Сейчас мы изложим эту конструкцию (хотя детали построения будут иными).

Определим последовательность функций $\alpha_0, \alpha_1, \ldots$ от одного аргумента. (Все эти функции будут всюду определёнными.) Положим $\alpha_0(x) = x + 1$. Определяя α_i, мы будем использовать такое обозначение: $f^{[n]}(x)$ означает $f(f(\ldots f(x)\ldots))$, где функция f использована n раз. Так вот,

$$\alpha_i(x) = \alpha_{i-1}^{[x+2]}(x)$$

(почему удобно применять функцию α_{i-1} ровно $x+2$ раза, мы увидим чуть позже).

Очевидные свойства (формально их можно доказать по индукции):

- $\alpha_i(x) > x$ при всех i и x;
- $\alpha_i(x)$ возрастает с возрастанием x;
- $\alpha_i(x)$ возрастает с возрастанием i (для каждого фиксированного x);
- $\alpha_i(x) \geqslant \alpha_{i-1}(\alpha_{i-1}(x))$.

Теперь можно оценить скорость роста любой примитивно рекурсивной функции.

Теорема 80. *Пусть f — примитивно рекурсивная функция n аргументов. Тогда найдётся такое k, что*

$$f(x_1, \ldots, x_n) \leqslant \alpha_k(\max(x_1, \ldots, x_n))$$

при всех $x_1, \ldots, x_n$.

Доказательство. Идея проста — можно оценить скорость роста композиции функций, зная оценки для каждой из них; аналогично для рекурсии. Формально говоря, доказательство использует "индукцию по построению" примитивно рекурсивных функций.

Для базисных функций утверждение очевидно. Посмотрим на подстановку. Пусть

$$f(x) = g(h_1(x), \ldots, h_k(x))$$

(для краткости мы пишем одну букву x, имея в виду вектор переменных). Пусть α_N оценивает все функции $h_1, \ldots, h_k$ и функцию g сверху, то есть $h_i(x) \leqslant \alpha_N(\max(x))$ при всех i и x, а также $g(y) \leqslant \alpha_N(\max(y))$ (здесь $\max(u)$ означает максимальный элемент в наборе u). Тогда $f(x)$ не превосходит

$$\alpha_N(\max(h_1(x), \ldots, h_k(x))) \leqslant \alpha_N(\alpha_N(x)) \leqslant \alpha_{N+1}(x)$$

(мы пользуемся указанными выше свойствами функций α_i).

Похоже (но немного сложнее) дело обстоит с рекурсией. Пусть функция f определяется рекурсивно:

$$f(x, 0) = g(x); \tag{1}$$

$$f(x, n+1) = h(x, n, f(x, n)). \tag{2}$$

(Здесь x также обозначает набор нескольких переменных.) Пусть функции g и h оцениваются сверху функцией α_N. Тогда

$$f(x,1) = h(x,0,f(x,0)) \leqslant \alpha_N(\max(x,0,f(x,0))) \leqslant \\ \leqslant \alpha_N(\max(x,0,\alpha_N(\max(x)))) \leqslant \alpha_N(\alpha_N(\max(x))) \tag{3}$$

(в последнем переходе мы пользуемся тем, что $\alpha_N(t) > t$). Аналогично $f(x,2) \leqslant \alpha_N(\alpha_N(\alpha_N(\max(x))))$ и вообще

$$f(x,i) \leqslant \alpha_N^{[i+1]}(\max(x)) \leqslant \alpha_{N+1}(\max(i, \max(x))),$$

что и требовалось доказать. □

Заметим, что каждый оператор подстановки или рекурсии увеличивает номер верхней оценки на 1, так что функция, в определении которой не более 100 операторов, растёт не быстрее α_{101}.

Очевидным следствием полученной оценки является такое утверждение:

Теорема 81. *Функция $A(n) = \alpha_n(n)$ растёт быстрее любой примитивно рекурсивной функции.*

Отметим, что определение функции Аккермана (точнее, функции $\langle n, x\rangle \mapsto \alpha_n(x)$) вполне можно назвать рекурсивным — одно значение этой функции определяется через другие, с меньшим первым аргументом. Оно является примером рекурсивного определения, не сводящегося к примитивной рекурсии.

Задача 196. *Покажите, что прямой пересчёт (в возрастающем порядке) бесконечного примитивно рекурсивного множества может не быть примитивно рекурсивным.*

Задача 197. *Покажите, что функция, обратная к примитивно рекурсивной биекции* $i\colon \mathbb{N} \to \mathbb{N}$, *может не быть примитивно рекурсивной.*

Задачи

§1. Множества и отображения

1.1. Множества

Основным типом примеров следующего пункта является “Доказать равенство множеств, заданных формулами алгебры множеств”. Рекомендуется перейти к формулам алгебры предикатов, определяющим эти множества, и вычислить, равносильны ли они, или, оставаясь в формулах алгебры множеств, перейти к булевым формулам алгебры множеств и воспользоваться основными равенствами булевой алгебры множеств.

Пример 1. Доказать, что $A\backslash(B \cup C) = (A\backslash B) \cap (A\backslash C)$.

Решение. Перейдём к булевым формулам алгебры множеств

$$A\backslash(B \cup C) = A \cap \overline{(B \cup C)} = A \cap (\overline{B} \cap \overline{C}) = \\ = (A \cap \overline{B}) \cap (A \cap \overline{C}) = (A\backslash B) \cap (A\backslash C).$$

Особое внимание следует уделить решению примеров, содержащих семейства множеств, так как операции над семействами множеств вводятся с помощью кванторов.

Пример 2. Доказать, что

$$A \cap \left(\bigcup_{i\in I} B_i\right) = \bigcup_{i\in I}(A \cap B_i).$$

Решение.

$$x \in A \cap \left(\bigcup_{i\in I} B_i\right) \equiv (x \in A) \wedge \left(x \in \bigcup_{i\in I} B_i\right) \equiv (x \in A) \wedge (\exists i(x \in B_i)) \equiv \\ \equiv \exists i((x \in A) \wedge (x \in B_i)) \equiv \exists i(x \in (A \cap B_i)) \equiv x \in \bigcup_{i\in I}(A \cap B_i).$$

1. Доказать, что множество A всех чётных чисел равно множеству B целых чисел, представимых в виде суммы двух нечётных целых чисел.

2. Доказать, что множество $A = \{x \mid x \in \mathbb{Z},\ x \text{ делится на } 6\}$ равно множеству $B = \{x \mid x \in \mathbb{Z},\ x \text{ делится на } 2,\ x \text{ делится на } 3\}$.

3. Доказать, что $\mathbb{Z} = \{x \mid \exists m \exists n (\subset \mathbb{Z})\ x = 3m + 5n\}$.

4. Привести пример таких множеств A, B, C, что $A \in B$, $B \in C$, но $A \notin C$.

5. Привести пример множеств A, B, таких, что $A \in B$ и $A \subset B$.

6. Доказать, что если $A_1 \subset A_2 \subset \ldots \subset A_n \subset A_1$, то $A_1 = A_2 = \ldots = A_n$.

7. Доказать, что $A \subset B$ тогда и только тогда, когда $A\backslash B = \varnothing$.

8. Доказать, что $A = B$ тогда и только тогда, когда $A \triangle B = \varnothing$.

Доказать равенства:

9. $A\backslash(B \cup C) = (A\backslash B) \cap (A\backslash C)$; **10.** $A\backslash(B \cap C) = (A\backslash B) \cup (A\backslash C)$;
11. $A\backslash(A\backslash B) = A \cap B$; **12.** $(A\backslash B)\backslash C = (A\backslash B)\backslash(B\backslash C)$; **13.** $A \triangle B = B \triangle A$;
14. $(A \triangle B) \triangle C = A \triangle (B \triangle C)$; **15.** $A \cap (B \triangle C) = (A \cap B) \triangle (A \cap C)$;
16. $A \triangle (A \triangle B) = B$.

17. Выразить операции $\cup$, $\backslash$ через $\triangle$, $\cap$.

18. Выразить операции $\cap$, $\backslash$ через $\triangle$, $\cup$.

19. Выразить операции $\cup$, $\cap$, через $\triangle$, $\backslash$.

20. Доказать, что нельзя выразить $\backslash$ через $\cup$ и $\cap$.

21. Доказать, что нельзя выразить $\cup$ через $\cap$ и $\backslash$.

22. Пусть $A = \{1; 4; 5\}$, $B = \{2; 4; 6\}$. Найти $A \cup B$, $A \cap B$, $A\backslash B$, $B\backslash A$, $A \triangle B$.

23. Перечислить все подмножества множества $\{1; 2; 3\}$, все собственные подмножества.

24. Доказать, что $2^{A\cap B} = 2^A \cap 2^B$, где 2^A — множество всех подмножеств множества A.

25. Пусть имеется последовательность множеств $A_1 \supset A_2 \supset \ldots A_n \supset \ldots$ Доказать, что $\bigcap\limits_{n\in\mathbb{N}} A_n = \bigcap\limits_{n_k\in\mathbb{N}} A_{n_k}$ для любой неограниченной последовательности натуральных чисел $\{n_k\}_{k=1}^{\infty}$.

Пусть $n\mathbb{Z}$ есть множество всех целых чисел, делящихся на n. Найти:

26. $n\mathbb{Z} \cap m\mathbb{Z}$; **27.** $\bigcup\limits_{n=2}^{\infty} n\mathbb{Z}$; **28.** $\bigcap\limits_{n=1}^{\infty} n\mathbb{Z}$; **29.** $\bigcup\limits_{p\in\mathbb{P}} p\mathbb{Z}$, где $\mathbb{P}$ — множество простых чисел; **30.** $\bigcup\limits_{n\in\mathbb{N}} \left[\frac{1}{n}; 1 - \frac{1}{n}\right]$; **31.** $\bigcap\limits_{n\in\mathbb{N}} \left[-\frac{1}{n}; 1 + \frac{1}{n}\right]$.

32. Пусть $C([a; b])$ — множество всех непрерывных функций, определённых на отрезке $[a; b]$,

$$C_x^3([a; b]) = \{f \in C([a; b]) \mid f(x) = 3\}.$$

Найти $\bigcup\limits_{x\in[a;b]} C_x^3([a; b])$, $\bigcap\limits_{x\in[a;b]} C_x^3([a; b])$.

Доказать:

33. $B \cap \Big(\bigcup\limits_{i\in I} A_i\Big) = \bigcup\limits_{i\in I}\Big(B \cap A_i\Big)$; **34.** $B \cup \Big(\bigcap\limits_{i\in I} A_i\Big) = \bigcap\limits_{i\in I}\Big(B \cup A_i\Big)$.

1.2. Отображения

Напомним, что отображение (функция) f действующее из X в Y, это тройка (X, Y, f), где X, Y — непустые множества, а f — правило, сопоста-

вляющее каждому элементу x из множества X элемент $f(x)$ из множества Y. Равенство отображений — это равенство троек.

Наибольшую сложность вызывают примеры на нахождение композиции отображений, заданных правилами, содержащими разветвления.

Пример 3. Пусть отображения $f, g : \mathbb{R} \to \mathbb{R}$ действуют по правилам:

$$f(x) = \begin{cases} x^3 & \text{при } |x| > 1, \\ -x & \text{при } |x| \leqslant 1; \end{cases}$$

$$g(x) = \begin{cases} x & \text{при } x > 8, \\ 2 - x & \text{при } |x| \leqslant 8, \\ 2 + x & \text{при } x < -8. \end{cases}$$

Найти $g \circ f$.

Решение. Композиция — это последовательное применение отображений. В нашем примере первым действует отображение f, вторым — g. Поэтому, нужно представить, что получится из области определения под действием отображения f, то есть множество $f(X)$. Полученное множество заданием g разбивается на части, область определения отображения f тоже разбивается на части.

Перепишем отображение f, убрав знак модуля:

$$f(x) = \begin{cases} x^3 & \text{при } x > 1, \\ -x & \text{при } | -1 \leqslant x \leqslant 1, \\ x^3 & \text{при } x < -1. \end{cases}$$

Если $x \in (1; \infty)$, то отображение f действует по правилу x^3 и множество $(1; \infty)$ отображается во множество $(1; \infty)$. На полученном множестве действие отображения g определяется как верхней, так и средней строкой. Чтобы чётко определить, когда какая строка действует, исходное множество разобьём точкой $x = 2$ на два подмножества: $(1; 2]$ и $(2; \infty)$. Тогда $f((1; 2]) = (1; 8]$ и $(1; 8]$ целиком попадает в среднюю строку определения отображения g, а $f((2; \infty)) = (8; \infty)$, что соответствует верхней строке определения g. Таким образом, мы получили, что

$$(g \circ f)(x) = \begin{cases} x^3 & \text{при } x \in (2; \infty), \\ 2 - x^3 & \text{при } x \in (1; 2]. \end{cases}$$

Если $x \in [-1; 1]$, то $f([-1; 1]) = [-1; 1]$, а это множество целиком попадает в среднюю строку определения g. Значит,

$$(g \circ f)(x) = 2 - (-x) = 2 + x \text{ при } x \in [-1; 1].$$

Если $x \in (-\infty; -1]$, то $f((-\infty; -1)) = (-\infty; -1)$. На этом множестве отображение g определяется как своей средней, так и нижней строкой. Разобьём множество $(-\infty; -1)$ на две части: $(-\infty; -2)$ и $[-2; -1)$. рассмотрим каждую из этих частей отдельно.

$f((-\infty; -2)) = (-\infty; -8)$. На этом множестве отображение g определяется своей нижней строкой, значит,

$$(g \circ f)(x) = 2 + x^3 \text{ при } x \in (-\infty; -2).$$

$f([-2; -1)) = [-8; -1)$. На этом множестве отображение g определяется своей средней строкой, значит,

$$(g \circ f)(x) = 2 - x^3 \text{ при } x \in [-2; -1).$$

Окончательно получаем

$$(g \circ f)(x) = \begin{cases} x^3 & \text{при } x \in (2; \infty), \\ 2 - x^3 & \text{при } x \in [-2; -1) \cup (1; 2], \\ 2 + x & \text{при } x \in [-1; 1], \\ 2 + x^3 & \text{при } x \in (-\infty; -2). \end{cases}$$

Пусть $f : X \to Y$ — произвольное отображение, B, B_1, B_2 — произвольные подмножества множества Y. Доказать, что:

35. $f^{-1}(B_1 \cup B_2) = f^{-1}(B_1) \cup f^{-1}(B_2)$; **36.** $f^{-1}(B_1 \cap B_2) = f^{-1}(B_1) \cap f^{-1}(B_2)$;

37. $f^{-1}(Y \backslash B) = X \backslash f^{-1}(B)$; **38.** $f^{-1}(B_1 \backslash B_2) = f^{-1}(B_1) \backslash f^{-1}(B_2)$;

39. $B_1 \subset B_2 \Rightarrow f^{-1}(B_1) \subset f^{-1}(B_2)$.

40. Привести пример, показывающий, что импликация

$$f^{-1}(B_1) \subset f^{-1}(B_2) \;\Rightarrow\; B_1 \subset B_2,$$

вообще говоря, не имеет места.

41. Доказать, что если $f : X \to Y$ и $A \subset X$, то

$$f(A) = \{y \in Y \mid \exists x (\in X)\ (x \in A) \wedge (y = f(x))\}.$$

Пусть $f : X \to Y$ — произвольное отображение, A_1, A_2 — произвольные подмножества множества X. Доказать, что:

42. $f(A_1 \cup A_2) = f(A_1) \cup f(A_2)$; **43.** $f(A_1 \cap A_2) \subset f(A_1) \cap f(A_2)$.

44. Привести пример, показывающий, что включение

$$f(A_1) \cap f(A_2) \subset f(A_1 \cap A_2),$$

вообще говоря, не имеет места.

45. Доказать, что

$$f(A_1) \backslash f(A_2) \subset f(A_1 \backslash A_2).$$

46. Привести пример, показывающий, что включение

$$f(A_1 \backslash A_2) \subset f(A_1) \backslash f(A_2),$$

вообще говоря, не имеет места.

47. Доказать, что

$$(A_1 \subset A_2) \;\Rightarrow\; (f(A_1) \subset f(A_2)).$$

48. Привести пример, показывающий, что включение

$$(f(A_1) \subset f(A_2)) \;\Rightarrow\; (A_1 \subset A_2),$$

вообще говоря, не имеет места.

Доказать, что для произвольного подмножества B области значений Y отображения $f : X \to Y$ выполняются соотношения:

49. $f(f^{-1}(B)) = B \cap f(X)$; **50.** $f^{-1}(B) = \varnothing \Leftrightarrow B \cap f(X) = \varnothing$.

51. Доказать, что для произвольного подмножества A области определения X отображения $f : X \to Y$ имеет место соотношение $A \subset f^{-1}(f(A))$.

Пусть $f : X \to Y$, $A \subset X$, $B \subset Y$. Доказать, что:

52. $f(A) \cap B = f(A \cap f^{-1}(B))$; **53.** $f(A) \cap B = \varnothing \Leftrightarrow A \cap f^{-1}(B) = \varnothing$;

54. $f(A) \subset B \Leftrightarrow A \subset f^{-1}(B)$.

Пусть $f : X \to Y$, $g : Y \to Z$, $A \subset X$, $C \subset Z$. Доказать, что:

55. $(g \circ f)^{-1}(C) = f^{-1}(g^{-1}(C))$; **56.** $(g \circ f)(A) = g(f(A))$.

57. Доказать, что если $A \subset X$, $B \subset X$, $i_A : A \to X$ — отображение включения, то $i_A^{-1}(B) = A \cap B$.

58. Пусть $f : X \to Y$, $A \subset X$, $B \subset Y$, $g = f|_A$: $A \to Y$ — сужение отображения f на A. Доказать, что $g^{-1}(B) = A \cap f^{-1}(B)$.

Доказать, что если $f : X \to Y$ — инъективное отображение, то для любых подмножеств A, A_1, A_2 его области определения X имеют место соотношения:

59. $f(A_1 \cap A_2) = f(A_1) \cap f(A_2)$; **60.** $f(A_1 \backslash A_2) = f(A_1) \backslash f(A_2)$;

61. $A_1 \subset A_2 \Leftrightarrow f(A_1) \subset f(A_2)$; **62.** $f^{-1}(f(A)) = A$.

63. Пусть для отображения $f : X \to X$ и для некоторого натурального числа n выполнено $f^n = e_X$. Доказать, что отображение f биективно.

64. Доказать, что если отображение $f \circ g$ инъективно, то g также инъективно.

65. Доказать, что если отображение $f \circ g$ сюръективно. то f также сюръективно.

66. Пусть заданы множества A, B, C, D и отображения $f : A \to B$, $g : B \to C$, $h : C \to D$. Доказать, что если отображения $g \circ f$ и $h \circ g$ биективны, то и отображения f, g, h биективны.

67. Пусть заданы множества A, B, C и отображения $f : A \to B$, $g : B \to$ $\to C$, $h : C \to A$. Доказать, что если среди отображений $h \circ g \circ f$, $g \circ f \circ h$,

$f \circ h \circ g$ два являются инъективными (сюръективными), а третье сюръективно (инъективно), то отображения f, g, h биективны.

Для следующих отображений $f, g : \mathbb{R} \to \mathbb{R}$ найти композиции $f \circ g$, $g \circ f$.

68.

$$f(x) = \begin{cases} 1+x & \text{при } x \geqslant 0, \\ 1-x & \text{при } x < 0; \end{cases} \qquad g(x) = \begin{cases} 1+x & \text{при } x \geqslant 1, \\ 2x & \text{при } x < 1; \end{cases}$$

69.

$$f(x) = \begin{cases} x^2 & \text{при } x \geqslant 1, \\ x & \text{при } x < 1; \end{cases} \qquad g(x) = \begin{cases} |x| & \text{при } x < 2, \\ 4-x & \text{при } x \geqslant 2. \end{cases}$$

70. Пусть отображение $f : \mathbb{R} \to \mathbb{R}$ действует по правилу

$$f(x) = \begin{cases} 1+x & \text{при } x \geqslant 0, \\ 1-x & \text{при } x < 0. \end{cases}$$

Найти $f([0;1])$, $f([-1;2])$, $f^{-1}([0;1])$, $f^{-1}([-1;2])$.

71. Пусть задано отображение $f : \mathbb{R} \to \mathbb{R}$, где $f(x) = \sin x$. Найти $f((0;\pi))$, $f\left(\left(\frac{\pi}{4};\frac{5\pi}{6}\right)\right)$, $f^{-1}\left(\left(-\frac{1}{2};\frac{1}{2}\right)\right)$, $f^{-1}([0;2])$.

72. Является ли отображение

$$f_n : \mathbb{N} \to \mathbb{N}, \qquad f_n(k) = \begin{cases} n-k & \text{при } k < n, \\ n+k & \text{при } k \geqslant n \end{cases}$$

инъективным, сюръективным, биективным?

Пусть $C(\mathbb{R})$ — множество всех вещественных непрерывных функций. Проверить, являются ли следующие отображения $F : C(\mathbb{R}) \to C(\mathbb{R})$ инъективным, сюръективным, биективным. Найти обратные к ним с соответствующей стороны.

73. $[F(f)](x) = f(e^x)$; **74.** $[F(f)](x) = e^{f(x)}$; **75.** $[F(f)](x) = (x^2-1)f(x)$; **76.** $[F(f)](x) = (x^2+1)f(x)$; **77.** $[F(f)](x) = f(2x-1)$; **78.** $[F(f)](x) = f^3(x)$; **79.** $[F(f)](x) = f(x^{\frac{1}{3}})$.

80. Найти композиицию отображений из задач 75, 76, 78 и 79.

§2. Алгебра высказываний

2.1. Таблицы истинности

Каждая формула алгебры высказываний обладает свойством превращаться в высказывание при фиксации в ней значений всех высказывательных переменных, то есть если мы зафиксируем в формуле значения всех

высказывательных переменных, то, пользуясь определениями логических операций, мы можем вычислить значение истинности формулы.

Таблица истинности формулы алгебры высказываний содержит столько строк, сколько всевозможных наборов значений истинности переменных можно образовать. Каждая высказывательная переменная может принимать только два значения (0 и 1), поэтому в случае n переменных таблица истинности содержит 2^n строк.

При построении таблицы истинности наборы значений переменных располагают сверху вниз в лексикографическом порядке (каждый набор понимают как двоичную запись неотрицательного целого числа и располагают в порядке возрастания от $(000\ldots0)$ до $(111\ldots1)$.

Пример 4. Построить таблицу истинности формулы:

$$x_1\overline{x}_2 \to (x_1 \vee x_2)\overline{x}_3.$$

Решение. 1. Определим порядок действий в формуле:

$$x_1 \overset{2}{\cdot} \overset{1}{\overline{x}}_2 \overset{6}{\to} (x_1 \overset{3}{\vee} x_2) \overset{5}{\cdot} \overset{4}{\overline{x}}_3 .$$

2. Пользуясь определениями операций $\neg$, $\cdot$, $\vee$ и $\to$, заполним таблицу.

x_1	x_2	x_3	$\overline{x}_2$	$x_1 \cdot \overline{x}_2$	$x_1 \vee x_2$	$\overline{x}_3$	$(x_1 \vee x_2)\overline{x}_3$	$x_1\overline{x}_2 \to (x_1 \vee x_2)\overline{x}_3$
0	0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0	1
0	1	0	0	0	1	1	1	1
0	1	1	0	0	1	0	0	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	0	0	1	0	0	1

Составить таблицы истинности для следующих формул:

81. $x \vee \overline{y}$; **82.** $x \wedge \overline{y}$; **83.** $x \to (y \vee x)$; **84.** $x \to (x \wedge y)$; **85.** $(x \vee y) \to (\overline{x} \vee \overline{y})$; **86.** $x \to ((x \vee y) \vee z)$; **87.** $x \to (y \to z)$; **88.** $(x \to y) \to z$; **89.** $x \sim (y \sim z)$; **90.** $(x \sim y) \sim z$; **91.** $(x \vee (y \vee z)) \to (\overline{x} \wedge (\overline{y} \wedge \overline{z}))$; **92.** $(x \to (y \wedge z)) \to (x \to (y \wedge z))$; **93.** $\left(x \sim \overline{(y \vee z)}\right) \sim (x \sim (y \vee z))$; **94.** $(x \vee \overline{y}) \to ((y \wedge \overline{z}) \to (x \vee (y \sim z)))$; **95.** $((x \sim y) \sim ((z \to (\overline{x} \vee \overline{y})) \to \to \overline{z})) \sim (x \vee y)$; **96.** $(x \sim y) \to (((y \sim z) \to (z \sim x)) \to (x \sim z))$.

Пусть x_i $(i = 1, 2, 3)$ — символы булевых переменных (то есть принимающих два значения: $0, 1$). Построить таблицы истинности:

97. $(x_1 = x_2) \vee (x_2 = x_3)$; **98.** $(x_1 > x_2) \to (x_2 = x_3)$; **99.** $(x_1 \neq x_2) \vee (x_2 \neq x_3)$; **100.** $((x_1 > x_2) \wedge (x_2 = x_3)) \to (x_1 > x_3)$.

Применяя таблицы истинности, доказать тождественную истинность формул:

101. $x \sim x$; **102.** $x \vee \overline{x}$; **103.** $\overline{(x \wedge \overline{x})}$; **104.** $\overline{\overline{x}} \sim x$;
105. $x \to (y \to x)$; **106.** $\overline{x} \to (x \to y)$; **107.** $((x \to y) \wedge x) \to y$;
108. $((x \to y) \wedge \overline{y}) \to \overline{x}$; **109.** $((x \vee y) \wedge \overline{x}) \to y$; **110.** $((x \sim y) \wedge \overline{x}) \to \overline{y}$;
111. $(x \to y) \sim (\overline{y} \to \overline{x})$; **112.** $((x \to y) \wedge (y \to z)) \to (x \to z)$;
113. $(x \to (y \to z)) \to ((x \wedge y) \to z)$; **114.** $((x \to z) \wedge (y \to z)) \to ((x \vee y) \to z)$;
115. $(x \to (y \to z)) \to ((x \to y) \to (x \to z))$.

Применяя таблицы истинности, доказать равносильность формул:

116. $x \vee y \equiv y \vee x$; **117.** $x \vee (y \vee z) \equiv (x \vee y) \vee z$; **118.** $x \wedge (y \vee z) \equiv (x \wedge y) \wedge z$;
119. $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$; **120.** $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$;
121. $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$;
122. $\overline{(x \vee y)} \equiv \overline{x} \wedge \overline{y}$
123. $\overline{(x \wedge y)} \equiv \overline{x} \vee \overline{y}$ — законы де Моргана;
124. $x \vee x \equiv x$
125. $x \wedge x \equiv x$ — законы идемпотентности;
126. $x \vee 0 \equiv x$; **127.** $x \wedge 1 \equiv x$; **128.** $\overline{\overline{x}} \equiv x$; **129.** $x \sim y \equiv y \sim x$;
130. $x \sim (y \sim z) \equiv (x \sim y) \sim z$; **131.** $x \to y \equiv \overline{x} \vee y$; **132.** $x \sim y \equiv (x \to$
$\to y) \wedge (y \to x)$.

2.2. Порядок действий и упрощённая запись формул

Учитывая соглашения о порядке выполнения операций, опустить "лишние" скобки и знак "$\wedge$" в формулах:

133. $x \wedge (y \wedge (\overline{x} \vee \overline{y}))$; **134.** $(x \wedge y) \vee ((y \wedge z) \vee ((\overline{x} \wedge y) \vee (x \wedge \overline{z})))$;
135. $((x \vee y) \vee z) \to ((x \wedge \overline{y}) \vee z)$; **136.** $((x \vee y) \wedge (x \vee (y \wedge z))) \to ((\overline{x} \wedge \overline{y}) \to \overline{z})$;
137. $((x \vee y) \vee (x \vee ((y \wedge (x \vee z)) \wedge (y \to z))) \sim \overline{z})$; **138.** $((x \vee y) \to$
$\to (x \wedge y)) \vee ((\overline{x} \wedge y) \vee (\overline{x} \vee y))$; **139.** $((x \vee y) \wedge z) \to (((x \vee \overline{y}) \vee z) \sim (\overline{x} \vee y))$;
140. $(x \wedge (y \vee z)) \wedge ((x \to (y \to z)) \sim (x \wedge y))$.

Восстановить скобки и знак "$\wedge$" в формулах:

141. $x \vee y \to z$; **142.** $x \vee y \to xy$; **143.** $\overline{xy} \vee x\overline{y}\overline{(y \vee z)}$;
144. $x \vee y(xy \vee z)$; **145.** $xy \vee x\overline{yz} \to \overline{x} \vee yz$; **146.** $(x \to x \vee yz) \sim (x \vee y \to z)$;
147. $(x \vee y)\overline{z} \to (xy \sim \overline{y} \vee \overline{z})$; **148.** $x \vee y \to x \vee y(x \to z) \vee x(y \sim z)$;
149. $xyz \to (x \sim yz) \vee x \vee y(x \to (y \sim z))$; **150.** $xy \sim x(y \to z)(x \sim y) \vee xz \vee yz$.

2.3. Равносильные преобразования и упрощение формул

Методом решения примеров на равносильные преобразования и упрощение формул является использование 19 основных равносильностей булевой алгебры высказываний, поэтому первым шагом при решении таких примеров является переход к булевым операциям с помощью формул:

$$a \to b \equiv \overline{a} \vee b,$$
$$a \sim b \equiv (a \to b)(b \to a) \equiv ab \vee \overline{a}\overline{b} \equiv (\overline{a} \vee b)(a \vee \overline{b}).$$

Следует иметь в виду, что буквы, использованные при записи основных равносильностей, могут означать как символы высказывательных переменных, так и формулы алгебры высказываний, то есть основная равносильность

$$a \vee \overline{a} \equiv 1$$

означает, в частности, что

$$x_1 \vee \overline{x}_1 \equiv 1,$$
$$1 \vee \overline{1} \equiv 1,$$
$$(x_1 \to x_2)\overline{x}_3 \vee \overline{(x_1 \to x_2)\overline{x}_3} \equiv 1.$$

Полезными при решении примеров на упрощение формул являются законы полупоглощения:

$$1)\ \ a \vee \overline{a}b \equiv a \vee b; \qquad 1')\ \ \overline{a} \vee ab \equiv \overline{a} \vee b;$$
$$2)\ \ a \cdot (\overline{a} \vee b) \equiv ab; \qquad 2')\ \ \overline{a}(a \vee b) \equiv \overline{a}b,$$

которые доказываются при помощи дистрибутивного закона:

$$a \vee \overline{a} \cdot b \equiv (a \vee \overline{a})(a \vee b) \equiv 1(a \vee b) \equiv a \vee b;$$
$$a(\overline{a} \vee b) \equiv a \cdot \overline{a} \vee a \cdot b \equiv 0 \vee ab \equiv ab.$$

Пример 5. С помощью равносильных преобразований упростить формулу

$$x_1\overline{x}_2 \to (\overline{x}_1 \vee \overline{x}_2)\overline{x}_3.$$

Решение.

$$x_1\overline{x}_2 \to (\overline{x}_1 \vee \overline{x}_2)\overline{x}_3 \underset{\text{булевым операциям}}{\overset{\text{переход к}}{\equiv}} \overline{x_1\overline{x}_2} \vee (\overline{x}_1 \vee \overline{x}_2)\overline{x}_3 \equiv$$

$$\underset{\text{дистрибутивный закон}}{\overset{\text{закон де Моргана}}{\equiv}} \overline{x}_1 \vee \overline{\overline{x}}_2 \vee \overline{x}_1\overline{x}_3 \vee \overline{x}_2\overline{x}_3 \underset{\text{двойного отрицания}}{\overset{\text{закон}}{\equiv}}$$

$$\equiv \underbrace{\overline{x}_1 \vee \overline{x}_1 x_3}_{\text{закон поглощения}} \vee \underbrace{x_2 \vee \overline{x}_2\overline{x}_3}_{\text{закон поглощения}} \equiv$$

$$\equiv \overline{x}_1 \vee x_2 \vee \overline{x}_3 \equiv \begin{cases} \text{а)} & \overline{x}_1 \vee (x_2 \vee \overline{x}_3) \equiv x_1 \to (x_2 \vee \overline{x}_3), \\ \text{б)} & \overline{x}_3 \vee (\overline{x}_1 \vee x_2) \equiv x_3 \to (x_1 \vee x_2), \\ \text{в)} & x_2 \vee \overline{x}_1 \vee \overline{x}_3 \equiv x_2 \vee \overline{x_1 x_3}, \\ \text{г)} & \overline{x}_1 \vee \overline{x}_3 \vee x_2 \equiv \overline{x_1 x_3} \vee x_2 \equiv x_1 x_3 \to x_2, \\ \text{д)} & \overline{x}_1 \vee x_2 \vee \overline{x}_3 \equiv \overline{x_1 \cdot \overline{x}_2 \cdot x_3}. \end{cases}$$

Замечание. Любую запись а) — д) можно считать ответом.

Следующий тип примеров — доказательство равносильности двух заданных формул с помощью равносильных преобразований. Существуют три основные схемы решения таких примеров. Каждая из них предполагает выполнение перехода к булевым операциям в исходных формулах.

Далее, по первой схеме предполагается, начиная с левой формулы, провести цепочку равносильных преобразований, завершив её на правой формуле.

Вторая схема — зеркальное отражение первой.

Третья схема предполагает проведение параллельных цепочек равносильных преобразований левой и правой формул до тех пор, пока в этих цепочках не обнаружится совпадение каких-то звеньев (одного звена левой цепочки с одним звеном правой).

Пример 6. Доказать, что

$$(x_1 \to x_3)(x_2 \to x_3) \equiv (x_1 \vee x_2) \to x_3.$$

Решение. Перейдём к булевым операциям

$$(\overline{x}_1 \vee x_3)(\overline{x}_2 \vee x_3) \equiv \overline{x_1 \vee x_2} \vee x_3.$$

1-я схема:

$$(\overline{x}_1 \vee x_3)(\overline{x}_2 \vee x_3) \underset{\text{закон}}{\overset{\text{дистрибутивный}}{\equiv}} \overline{x}_1\overline{x}_2 \vee \overbrace{x_3\overline{x}_2 \vee \underbrace{\overline{x}_1 x_3 \vee x_3}_{\text{закон поглощения}}}^{\text{закон поглощения}} \equiv$$

$$\equiv \overline{x}_1\overline{x}_2 \vee x_3 \underset{\text{де Моргана}}{\overset{\text{закон}}{\equiv}} \overline{x_1 \vee x_2} \vee x_3.$$

2-я схема:

$$\overline{x_1 \vee x_2} \vee x_3 \underset{\text{де Моргана}}{\overset{\text{закон}}{\equiv}} \overline{x}_1 \cdot \overline{x}_2 \vee x_3 \underset{\text{закон}}{\overset{\text{дистрибутивный}}{\equiv}} (\overline{x}_1 \vee x_3)(\overline{x}_2 \vee x_3).$$

3-я схема:

$$(\overline{x}_1 \vee x_2)(\overline{x}_2 \vee x_3) \overset{\text{дистрибутивный закон}}{\equiv} \overline{x}_1\overline{x}_2 \vee x_2\overline{x}_2 \vee \overline{x}_1 x_3 \vee x_3 \equiv \overline{x}_1\overline{x}_2 \vee x_3;$$

$$\overline{x_1 \vee x_2} \vee x_3 \overset{\text{закон де Моргана}}{\equiv} \overline{x}_1\overline{x}_2 \vee x_3.$$

Замечание. Следует иметь в виду, что среди примеров на доказательство равносильности формул есть примеры с отрицательным ответом. В этом случае ни одна из схем не приводит к получению ответа. Однако, неудача при использовании схем 1 — 3 может говорить и о недостаточно высокой технике равносильных преобразований. В случае неудачных попыток применения схем 1 — 3 следует для обеих формул построить таблицы истинности. Совпадение столбцов значений формул будет означать их равносильность, а несовпадение — неравносильность.

Применяя равносильные преобразования, доказать следующие соотношения:

151. $x \vee y \equiv \overline{\overline{x} \cdot \overline{y}}$; **152.** $\overline{xy} \equiv \overline{x} \vee \overline{y}$; **153.** $x \to y \equiv \overline{x \cdot \overline{y}}$; **154.** $x \to y \equiv \overline{y} \to \overline{x}$;
155. $xy \vee x\overline{y} \equiv x$; **156.** $x \vee xy \equiv x$; **157.** $x(x \vee y) \equiv x$; **158.** $x \vee \overline{x}y \equiv x \vee y$;
159. $x(\overline{x} \vee y) \equiv xy$; **160.** $(x \to y) \to y \equiv x \vee y$; **161.** $(x \vee y)(x \vee \overline{y}) \equiv x$;
162. $\overline{x} \vee \overline{y} \equiv y \to \overline{x}$; **163.** $x \sim y \equiv \overline{x} \sim \overline{y}$; **164.** $xy \vee \overline{x}y \vee \overline{xy} \equiv x \to y$;
165. $x \to (y \to z) \equiv (x \vee z)(y \vee z)$; **166.** $x \to (y \to z) \equiv y \to (x \to z)$;
167. $\overline{x} \vee xy \vee xz \vee \overline{x}y \vee \overline{x}z \equiv x \to y \vee z$.

Применяя равносильные преобразования, доказать тождественную истинность формул:

168. $x \to x \vee y$; **169.** $xy \to x$; **170.** $\overline{x} \to (x \to y)$; **171.** $(x \to y) \to (\overline{x} \vee y)$;
172. $(x \vee \overline{x}y) \sim (x \vee y)$; **173.** $(\overline{x} \to y) \to (\overline{y} \to x)$; **174.** $(\overline{x} \to \overline{y}) \to (y \to x)$;
175. $(x \to y) \vee (y \to x)$; **176.** $(x \to y) \vee (x \to \overline{y})$; **177.** $x \to (y \to xy)$;
178. $(x \to y)x \to y$; **179.** $(x \to y)\overline{y} \to \overline{x}$; **180.** $(x \vee y)\overline{x} \to y$;
181. $(x \vee\!\vee y)x \to \overline{y}$ (“$\vee\!\vee$” — альтернативная дизъюнкция: $(x \vee\!\vee y) \equiv \overline{x \sim y}$);
182. $(x \to y)(y \to z) \to (x \to z)$; **183.** $(x \to (y \to z)) \to (xy \to z)$;
184. $(x \to z)(y \to z) \to (x \vee y \to z)$; **185.** $(x \to z) \to ((y \to z) \to (x \vee y \to z))$.

Применяя равносильные преобразования, “упростить”:

186. $\overline{\overline{xy}} \vee (x \to y)x$; **187.** $\left(\overline{\overline{x} \vee y} \to x \vee y\right) y$; **188.** $\overline{(x \to y)(y \to \overline{x})}$;
189. $(x \vee y)(x \sim y)$; **190.** $(x \to y)(y \to z) \to (z \to x)$; **191.** $xz \vee x\overline{z} \vee yz \vee \overline{x}yz$;
192. $\overline{xy(x \to y)}$; **193.** $xy(x \sim y)$; **194.** $(x \to \overline{y})(x \sim y)$; **195.** $(x \to \overline{y}) \vee \overline{(x \vee y)}$.

Следующие формулы преобразовать так, чтобы они содержали только “$\wedge$” и “$\neg$”:

196. $x \vee y$; **197.** $x \to y$; **198.** $x \sim y$; **199.** $x \vee y \vee z$; **200.** $x \to (y \to z)$;

201. $x \vee (x \sim y)$; **202.** $\overline{x \to y} \vee (\overline{x} \to \overline{y})$; **203.** $x \vee\vee y$; **204.** $x\overline{y} \to (\overline{y} \to x)$; **205.** $x \vee y \to (\overline{x} \to z)$.

Следующие формулы преобразовать так, чтобы они содержали только "$\vee$" и "$\neg$":

206. xy; **207.** xyz; **208.** $x \sim y$; **209.** $x \vee\vee y$; **210.** $x(y \sim z)$; **211.** $x \sim y \sim z$; **212.** $(x \sim y)(y \sim z)$; **213.** $xy \sim xz$.

Преобразовать следующие формулы так, чтобы знак отрицания был отнесён только к переменным высказываниям:

214. $\overline{\overline{x} \vee y}$; **215.** $\overline{xy \vee z}$; **216.** $\overline{xy \vee \overline{z}} \to \overline{xyz}$; **217.** $\overline{x \to (y \to z)}$; **218.** $\overline{x \to y \to (\overline{x} \to \overline{z})}$; **219.** $\overline{(x \sim y)(y \sim z)}$.

Преобразовать формулы так, чтобы они содержали только операции "$\vee$", "$\wedge$" и "$\neg$":

220. $x \sim y$; **221.** $(x \to y) \sim (y \to z)$; **222.** $(x \sim y) \to (y \to z)$; **223.** $(x \sim y) \to (y \sim z)$; **224.** $(x \sim y)(y \sim z) \to (x \sim z)$; **225.** $(x \sim y) \vee (y \sim z) \to (x \sim y \sim z)$; **226.** $x \sim y \sim z \sim v$; **227.** $(x \to y) \sim (z \to (x \sim \overline{z}))$.

§3. Двойственность в алгебре высказываний

Построение двойственных формул основано на общем и булевом принципах двойственности.

Общий принцип двойственности состоит в следующем: если исходная формула представляет собой подстановку формул в формулу, то двойственная формула — аналогичная подстановка двойственных формул в двойственную формулу.

Пример 7. Пусть

$$F(x_1, x_2, x_3) = (x_1 \to \overline{x}_2 x_3) \vee (x_1 \sim x_3).$$

Найти двойственную формулу F^*.

Решение. Обозначим $y_1 = x_1 \to \overline{x}_2 x_3$, $y_2 = x_1 \sim x_3$, тогда $F = y_1 \vee y_2$. Найдём двойственные формулы для подставляемых формул (y_1, y_2) и для

формулы, в которую осуществляется подстановка (F):

$$(y_1 \vee y_2)^* \equiv \overline{\overline{y}_1 \vee \overline{y}_2} \equiv \overline{\overline{y}}_1 \cdot \overline{\overline{y}}_2 \equiv y_1 \cdot y_2;$$

$$(x_1 \to \overline{x}_2 x_3)^* \equiv \overline{\overline{x}_1 \to \overline{\overline{x}}_2 \overline{x}_3} \equiv \overline{\overline{\overline{x}}_1 \vee x_2 \overline{x}_3} \equiv$$

$$\equiv \overline{x_1 \vee x_2 \overline{x}_3} \equiv \overline{x}_1 \cdot \overline{x_2 \cdot \overline{x}_3} \equiv \overline{x}_1(\overline{x}_2 \vee x_3) \equiv \overline{x}_1(x_2 \to x_3);$$

$$(x_1 \sim x_3)^* \equiv \overline{\overline{x}_1 \sim \overline{x}_3} \equiv \overline{\overline{x}_1 \cdot \overline{x}_3 \vee \overline{\overline{x}}_1 \cdot \overline{\overline{x}}_3} \equiv \overline{\overline{x}_1 \overline{x}_3 \vee x_1 x_3} \equiv$$

$$\equiv \overline{\overline{x}_1 \overline{x}_3} \cdot \overline{x_1 \cdot x_3} \equiv (x_1 \vee x_3)(\overline{x}_1 \vee \overline{x}_3) \equiv \overline{x}_1 x_3 \vee x_1 \overline{x}_3 \equiv x_1 \sim \overline{x}_3.$$

Применим теперь общий принцип двойственности:

$$F^* \equiv y_1 \cdot y_2 \equiv \overline{x}_1(x_2 \to x_3)(x_1 \sim \overline{x}_3).$$

Булев принцип двойственности состоит в следующем: формула, двойственная к булевой формуле получается заменой $\vee$ на $\wedge$, $\wedge$ на $\vee$, 0 на 1, 1 на 0 и сохранением структуры формулы.

Пример 8. Найти формулу, двойственную к формуле

$$(x_1 \to \overline{x}_2 x_3) \vee (x_1 \sim x_3),$$

пользуясь булевым принципом двойственности.

Решение.

$$((x_1 \to \overline{x}_2 x_3) \vee (x_1 \sim x_3))^* \equiv ((\overline{x}_1 \vee \overline{x}_2 x_3) \vee (x_1 x_3 \vee \overline{x}_1 \overline{x}_3))^* \overset{\text{булев принцип двойственности}}{\equiv}$$

$$\equiv \overline{x}_1 \cdot (\overline{x}_2 \vee x_3) \cdot ((x_1 \vee x_3)(\overline{x}_1 \vee \overline{x}_3)) \overset{\text{дистрибутивный закон}}{\equiv}$$

$$\equiv \overline{x}_1(x_2 \to x_3)(x_1 \overline{x}_3 \vee x_3 \overline{x}_1) \equiv \overline{x}_1(x_2 \to x_3)(x_1 \sim \overline{x}_3).$$

Найти двойственные формулы:

228. $x(\overline{y} \vee z)$; **229.** $xy \vee xz$; **230.** $\overline{(x \vee y)}(x \vee \overline{yz})$; **231.** $(xy \vee yz \vee zv)\overline{(x \vee y \vee z)}$; **232.** $x\left(y \vee z\overline{(x \vee y)}\right)$; **233.** $\overline{xyz} \vee xy\overline{z} \vee x\overline{y}z \vee \overline{x}yz$; **234.** $\left((x \vee y)\overline{(x \vee z)} \vee xy\right) \vee$ $\vee \left(\overline{(x \vee y)z} \vee x\right)$; **235.** $xy\left(\overline{yz} \vee xyz\overline{(xz \vee yz)} \vee \overline{xy}\right)(x \vee y \vee z)$.

Применить закон двойственности к следующим равносильностям:

236. $xx \equiv x$; **237.** $x \vee 0 \equiv x$; **238.** $xy \equiv yx$; **239.** $x \vee (y \vee z) \equiv (x \vee y) \vee z$; **240.** $\overline{xy} \equiv \overline{x} \vee \overline{y}$; **241.** $x(x \vee y) \equiv x$; **242.** $x \vee \overline{x}y \equiv x \vee y$; **243.** $x \vee xy \vee yz \vee \overline{x}z \equiv x \vee z$.

§4. Нормальные формы: ДНФ, КНФ

Дизъюнктивную нормальную форму (ДНФ) можно построить, используя таблицу истинности или следующий алгоритм:

1. Перейти к булевым операциям.

2. Перейти к формуле с тесными отрицаниями, то есть к формуле, в которой отрицания находятся не выше, чем над переменными.

3. Раскрыть скобки.

4. Повторяющиеся слагаемые взять по одному разу.

5. Опустить тождественно ложные слагаемые, то есть слагаемые вида: $\ldots \cdot x_i \cdot \overline{x}_i \cdot \ldots$

6. Пополнить оставшиеся слагаемые недостающими переменными.
(Пример на пополнение (переменные x_1, x_2, x_3):
$\ldots \vee x_1\overline{x}_3 \vee \ldots \equiv \ldots x_1(x_2 \vee \overline{x}_2)\overline{x}_3 \ldots \equiv \ldots \vee x_1x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3 \vee \ldots$)

7. Повторяющиеся слагаемые взять по одному разу.

Пример 9. Найти ДНФ формулы

$$(x_1 \to x_2\overline{x}_3) \to (x_1 \sim x_3).$$

Решение.

$$(x_1 \to x_2\overline{x}_3)(x_1 \sim x_3) \overset{1}{\equiv} \overline{\overline{x}_1 \to x_2\overline{x}_3} \vee (x_1x_3 \vee \overline{x}_1\overline{x}_3) \overset{2}{\equiv}$$

$$\equiv x_1 \cdot (\overline{x}_2 \vee x_3) \vee x_1x_3 \vee \overline{x}_1\overline{x}_3 \overset{3}{\equiv}$$

$$\equiv x_1\overline{x}_2 \vee x_1x_3 \vee x_1x_3 \vee \overline{x}_1\overline{x}_3 \overset{4}{\equiv} x_1\overline{x}_2 \vee x_1x_3 \vee \overline{x}_1\overline{x}_3 \overset{6}{\equiv}$$

$$\equiv x_1\overline{x}_2x_3 \vee x_1\overline{x}_2\overline{x}_3 \vee x_1x_2x_3 \vee x_1\overline{x}_2x_3 \vee \overline{x}_1x_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2\overline{x}_3 \overset{7}{\equiv}$$

$$\equiv x_1\overline{x}_2x_3 \vee x_1\overline{x}_2\overline{x}_3 \vee x_1x_2x_3 \vee \overline{x}_1x_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2\overline{x}_3.$$

Совершенную конъюнктивную нормальную форму (КНФ) можно построить по следующей схеме:

$$f \equiv (f^*)^* \equiv (\text{ДНФ}(f^*))^* \underset{\text{двойственности}}{\overset{\text{принцип}}{\equiv}} \text{КНФ}(f).$$

Пример 10. Найти КНФ формулы

$$(x_1 \to x_2\overline{x}_3) \to (x_1 \sim x_3).$$

Решение.

$$(x_1 \to x_2\overline{x}_3) \to (x_1 \sim x_3) \equiv \left(\left(\overline{(x_1 \to x_2\overline{x}_3)} \vee (x_1x_3 \vee \overline{x}_1\overline{x}_3)\right)^*\right)^* \equiv$$

$$\equiv (x_1(\overline{x}_2 \vee x_3) \vee x_1x_3 \vee \overline{x}_1\overline{x}_3)^*)^* \equiv ((x_1 \vee \overline{x}_2x_3) \cdot (x_1 \vee x_3) \cdot (\overline{x}_1 \vee \overline{x}_3))^* \equiv$$

$$\equiv ((x_1 \vee x_1\overline{x}_2x_3 \vee x_1x_3 \vee \overline{x}_2x_3)(\overline{x}_1 \vee \overline{x}_3))^* \equiv (\overline{x}_1\overline{x}_2x_3 \vee x_1\overline{x}_3)^* \equiv$$

$$\equiv (\overline{x}_1\overline{x}_2x_3 \vee x_1x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3)^* \equiv (\overline{x}_1 \vee \overline{x}_2 \vee x_3)(x_1 \vee x_2 \vee \overline{x}_3)(x_1 \vee \overline{x}_2 \vee \overline{x}_3).$$

Известно, что ДНФ и КНФ определены формулой однозначно и, значит, их можно строить по таблице истинности формулы.

Схема построения ДНФ и КНФ по таблице истинности приведена ниже для формулы $(x_1 \to x_2\overline{x}_3)(x_1 \sim x_3)$.

x_1	x_2	x_3	$(x_1 \to x_2\overline{x}_3)(x_1 \sim x_3)$		
0	0	0	1	$\to$	$\overline{x}_1\overline{x}_2\overline{x}_3$
0	0	1	0	$\to$	$x_1 \vee x_2 \vee \overline{x}_3$
0	1	0	1	$\to$	$\overline{x}_1 x_2 \overline{x}_3$
0	1	1	0	$\to$	$x_1 \vee \overline{x}_2 \vee \overline{x}_3$
1	0	0	1	$\to$	$x_1\overline{x}_2\overline{x}_3$
1	0	1	1	$\to$	$x_1\overline{x}_2 x_3$
1	1	0	0	$\to$	$\overline{x}_1 \vee \overline{x}_2 \vee x_3$
1	1	1	1	$\to$	$x_1x_2x_3$

ДНФ: $\overline{x}_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3 \vee x_1\overline{x}_2x_3 \vee x_1x_2x_3$;

КНФ: $(x_1 \vee x_2 \vee \overline{x}_3)(x_1 \vee \overline{x}_2 \vee \overline{x}_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)$.

Привести к дизъюнктивной нормальной форме (ДНФ):

244. $x \to (y \to z)$; **245.** $\overline{xy} \vee (x \to y)$; **246.** $(x \vee y \vee z)(x \to y)$;
247. $(x \vee y)(y \vee z) \to (x \vee z)$; **248.** $x \sim y$; **249.** $x \vee \vee y$; **250.** $x \sim y \sim z$;
251. $(x \to y) \sim \overline{(x \to (y \to z))}$; **252.** $(x \sim y)(y \sim z) \to (x \sim z)$;
253. $(x \sim y)(y \sim z)(z \sim x)$.

Привести к конъюнктивной нормальной форме (КНФ):

254. $x \vee yz$; **255.** $xy \vee yz \vee \overline{z}$; **256.** $x \vee yz \vee \overline{x}\,\overline{y}\,\overline{z}$; **257.** $x \to yz$;
258. $x \to yzv$; **259.** $x \sim yz$; **260.** $xy \sim \overline{x}\,\overline{y}$; **261.** $x \sim y \sim z$;
262. $x \vee y \sim x \sim z$; **263.** $x \vee \vee (y \vee \vee z)$.

Приведением к нормальной форме выяснить, какие из формул являются тождественно истинными, тождественно ложными, выполнимыми:

264. $xy \to x \vee y$; **265.** $x \vee y \to xy$; **266.** $\overline{x}y \to x\overline{y}$;
267. $(x \to y)x \to x \vee y \vee z$; **268.** $x \vee y \to x \vee z$; **269.** $(x \to y) \to (\overline{y} \to \overline{x})$;
270. $(x \to z) \to ((y \to z) \to ((x \vee y) \to z))$; **271.** $\overline{x}yz \vee x\overline{y}z \vee xy\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z}$;
272. $xy \vee \overline{x}\,\overline{y} \sim (x \vee y)(\overline{x} \vee \overline{y})$.

Для каждой из следующих формул найти дизъюнктивное и конъюнктивное разложение:

273. $x \vee y$; **274.** xy; **275.** $x \to y$; **276.** $x \sim y$; **277.** $x \vee \vee y$;
278. $x \to (y \to x)$; **279.** $\overline{x}y\overline{(x \to y)}$; **280.** $x \vee y \to z$; **281.** $xy \to z$.

Привести к ДНФ следующие формулы:

282. $\overline{x} \vee \overline{y}$; **283.** $(\overline{x} \to y) \to x$; **284.** $x \to (y \to x)$; **285.** $x \to (y \to z)$;
286. $(x \to y)(y \to z) \to (x \to z)$; **287.** $(x \to y)(y \to z)(z \to x)$;
288. $(x \vee y)(y \vee z)(z \sim x)$; **289.** $(x \to y)(y \to z)(z \to v)$.

Привести к КНФ следующие формулы:
290. $(x \to y) \to x \vee \overline{y}$; **291.** $x\overline{x} \cdot \overline{y}$; **292.** $x\overline{y}(x \to y)$; **293.** $x \to yz$; **294.** xyz; **295.** $(x \vee y)(y \to z)(z \sim x)$; **296.** $x \vee y \to (x \to z)$; **297.** $((x \to y) \sim (y \to \overline{x}))z$; **298.** $x \vee y \vee z \to (x \vee y)z$; **299.** $xy \to zv$.

Приведением к нормальным формам доказать неравносильность следующих формул:
300. $x \vee y$ и $x \to y$; **301.** $x \to y$ и $x \sim y$; **302.** $x \vee y$ и $x \oplus y$; **303.** $x \to (y \to z)$ и $(x \to y) \to z$; **304.** $xy \vee z$ и $x(y \vee z)$; **305.** $(x \to y) \vee z$ и $x \vee y \to z$; **306.** $(x \to y)z$ и $x \to yz$; **307.** $(x \to y) \sim z$ и $(x \sim y) \to z$; **308.** $(x \vee y) \sim z$ и $(x \sim y) \vee z$; **309.** $xy \sim z$ и $(x \sim y)z$.

Следующие формулы разложить по переменным x, y, z:
310. xy; **311.** $x \vee y$; **312.** x; **313.** $(x \vee y)(\overline{x} \vee \overline{y})$; **314.** $xy \vee \overline{x}y \vee \overline{x}\,\overline{y}$.

Определение. Формула F называется логическим следствием формул (посылок) $f_1, \ldots, f_n$, если $f_1 \cdot f_2 \cdot \ldots \cdot f_n \to F \equiv 1$.

Выяснить, является ли первая формула логическим следствием остальных:
315. y; $x \to y$, x; **316.** x; $x \to y$, y; **317.** $\overline{x}$; $x \to y$, $\overline{y}$; **318.** $\overline{y}$; $x \to y$, $\overline{x}$; **319.** y; $x \vee y$, $\overline{x}$; **320.** y; $x \vee \vee y$, x; **321.** $x \to z$; $x \to y$, $y \to z$; **322.** $x \vee y \to z$; $x \to z$, $y \to z$; **323.** $z \to x$; $x \to y$, $\overline{y} \to \overline{z}$; **324.** $x \vee y$; $x \to y$, $\overline{y} \to \overline{x}$, $\overline{x} \vee \overline{y}$; **325.** $\overline{x}$; $x \sim y$, $y \vee \overline{z}$, z; **326.** z; $x \to y$, $\overline{y} \vee z$, x; **327.** $\overline{y} \vee \overline{z}$; $x \vee \overline{z}$, $y \to x \cdot z$, x; **328.** $z \to y$; $x \to y$, $\overline{x}$, z; **329.** $\overline{z} \to \overline{x}$; $x \to y$, xy, $\overline{z} \to \overline{y}$; **330.** $x \vee t$; $x \to y$, $y \to \overline{z}$, $x \vee z \to yt$; **331.** xt; $x \to z$, $\overline{y} \vee z$, $z \to y \vee t$, $z \vee t$.

Найти все (с точностью до равносильности) логические следствия из посылок:
332. x, $x \to y$; **333.** $\overline{x}$, $x \sim y$; **334.** x, $\overline{y}$, $x \vee y$; **335.** $x \to (y \to z)$, $y \to z$; **336.** $x \to (y \to z)$, $y \to \overline{z}$; **337.** $x \to y$, $y \to z$; **338.** $x \vee y$, $y \vee z$, $z \vee x$; **339.** x, $x \vee y$, $x \vee y \vee z$; **340.** $x \to (y \to (z \to t))$, $x \to (y \to z)$; **341.** $x \to (y \to z)$, $y \to (z \to t)$.

Найти все (с точностью до равносильности) посылки, логическим следствием которых являются формулы:
342. $x \cdot y$; **343.** $x \sim y$; **344.** $x \vee y$; **345.** $x \to y$; **346.** $x \vee y \to x \cdot y$; **347.** $x \cdot y \cdot z$; **348.** $(x \vee y) \cdot z$; **349.** $(x \to y) \cdot z$; **350.** $x \to y \cdot z$; **351.** $x \to (y \to \overline{z})$.

Определение. Вывод $f_1, \ldots, f_n \vdash F$ называется правильным, если формула F является логическим следствием формул $f_1, \ldots, f_n$.

Докажите правильность выводов:
352. $a \to b, a \vdash b$; **353.** $a \to b, \overline{b} \vdash \overline{a}$; **354.** $a \vee b, \overline{a} \vdash b$; **355.** $a \vee \vee b, a \vdash \overline{b}$;

356. $a \vee \vee b, \overline{a} \vdash b$; **357.** $a \to b, b \to c \vdash a \to c$; **358.** $a \vee b, a \to b \vdash b$;
359. $a \to b, b \to c, \overline{c} \vdash \overline{a}$; **360.** $a \to b, b \to c, a \vdash b$; **361.** $a \vee \vee b, a \to b \vdash b$;
362. $a \vee \vee b, b \vee \vee c \vdash a \to c$; **363.** $a \to b, b \to c, c \to a \vdash a \to bc$.

Выяснить, правильны ли следующие выводы:
364. $a \to b, b \vdash a$; **365.** $a \to b, \overline{a} \vdash \overline{b}$; **366.** $a \to b, \overline{a} \to \overline{b} \vdash a \sim b$;
367. $a \to b, \overline{b} \to \overline{a} \vdash a \sim b$; **368.** $a \to b, a \vee b \vdash a$; **369.** $a \to b$, $b \to a, a \vee b \vdash a \cdot b$; **370.** $a \to (b \to c), (a \to b) \to c \vdash b \to c$;
371. $a \to (b \to c), (a \to b) \to c \vdash a \to c$; **372.** $a \to bc, b \to ac$, $c \to ab, a \vee b \vee c \vdash a \cdot b \cdot c$; **373.** $a \vee b \to c, a \vee c \to b, b \vee c \to a, a \vee b \vee c \vdash a \cdot b \cdot c$.

§5. Функции алгебры логики

Напомним, что булевы операции $\neg, \wedge, \vee$ образуют полную систему функций. Это означает, что любая функция алгебры логики ($\Leftrightarrow$ булева функция) может быть задана формулой над $\neg, \wedge, \vee$.

В частности, $x \mid y \equiv \overline{x \cdot y}$, $x \uparrow y \equiv \overline{x \vee y}$, $x \oplus y \equiv x\overline{y} \vee \overline{x}y$.

Ещё одной полной системой функций является $\{0, 1, \oplus, \wedge\}$. Формулы над $\{0, 1, \oplus, \wedge\}$ называют многочленами Жегалкина.

Каноническим многочленом Жегалкина называют многочлен Жегалкина, в котором раскрыты скобки и приведены подобные члены.

Ниже используются следующие обозначения:

P_2 — все булевы функции;

P_0 — булевы функции, сохраняющие 0;

P_1 — булевы функции, сохраняющие 1;

S — самодвойственные булевы функции.

Если K — один из этих классов, то $K(n)$ обозначает булевы функции этого класса от n аргументов.

Переменная x_i функции называется фиктивной, если

$$f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n).$$

Переменная x_i в функции $f(x_1, \ldots, x_n)$ фиктивна тогда и только тогда, когда канонический многочлен Жегалкина функции f не содержит переменной x_i.

374. Найти канонические многочлены Жегалкина следующих булевых функций:
а) всех булевых функций из $P_2(1)$, $P_2(2)$; **б)** $(x_1 \to x_2) \sim (x_2 \sim x_3)$;
в) $(x_1 \to x_3) \cdot (x_2 \oplus x_3)$; **г)** $\overline{x_1 \cdot x_3} \vee x_2 \cdot \overline{x}_4$; **д)** $(x_1 \sim x_2) \to x_3$;
е) (10101100) — столбец значений функции f в её таблице; **ж)** (11000100);
з) $(\overline{x}_1 \mid x_2) \uparrow x_3$.

375. Найти все фиктивные переменные следующих булевых функций:
а) $x_1x_2 \vee x_1\overline{x}_2$; **б)** $x_1\overline{x}_2 \vee x_2$; **в)** $x_1\overline{x}_2 \vee x_1$; **г)** $(x_1 \to (x_2 \to x_3)) \to ((x_1 \to x_2) \to (x_1 \to x_3))$; **д)** $(x_1 \to x_2)((x_2 \to x_3) \to (x_1 \to x_3))$; **е)** $(x_1 \to x_2) \to (\overline{x}_2 \to \overline{x}_1)$; **ж)** $(x_1 \to x_2) \to x_1$.

376. Сколько функций содержится во множестве:
а) $P_0(n) \cap P_1(n)$; **б)** $P_0(n) \cup P_1(n)$; **в)** $P_0(n) \backslash P_1(n)$; **г)** $P_0(n) \cap S(n)$; **д)** $P_0(n) \cup S(n)$; **е)** $P_0(n) \backslash S(n)$; **ж)** $S(n) \backslash P_0(n)$.

377. Среди функций примеров 374 и 375 найти все функции, входящие:
а) в P_0; **б)** в P_1.

378. Какие из следующих функций самодвойственны:
а) $(x_1 \to x_2) \to x_1x_3$; **б)** $(\overline{x}_1 \vee x_2 \vee \overline{x}_1)x_4 \vee \overline{x}_1x_2\overline{x}_3$; **в)** $x_1x_2 \oplus x_1x_3 \oplus x_2x_3$; **г)** (0001001001100111); **д)** $f(x_1, x_2, \ldots, x_{2m+1}) = x_1 \oplus x_2 \oplus \ldots \oplus x_{2m+1} \oplus \delta$, $\delta \in \{0, 1\}$; **е)** $(x_1 \vee x_2)(x_1 \vee x_3)(x_2 \vee x_3)$; **ж)** $(x_1 \mid \overline{x}_1) \uparrow x_2$.

379. Из несамодвойственной функции f с помощью отождествления переменных и операции $\neg$ получить константу:
а) (00111001); **б)** $(x_1 \mid x_2) \to (x_1 \oplus x_3)$; **в)** $(x_1 \vee \overline{x}_2 \vee x_3) \oplus \overline{x}_1x_2x_3$; **г)** $x_1x_2 \vee x_1x_3 \vee x_2x_4 \vee x_3x_4$.

380. Какие из функций примеров 374, 375, 378 монотонны?

381. Из немонотонных функций примеров 378 и 379 с помощью подстановки констант получить $\neg x$.

382. Какие из следующих функций монотонны:
а) $x_1 \to (x_2 \to x_3)$; **б)** (00110111); **в)** $x_1x_3 \cdot (x_2 \oplus x_3)$; **г)** $x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1$; **д)** (01100111).

383. Какие из функций примеров 374, 375, 378, 382 линейны?

384. Из нелинейных функций примера 383 с помощью констант 0, 1 и операции $\neg$ получить $\wedge$.

385. Выразить с помощью суперпозиций:
а) $\wedge$ и $\to$ через $\neg, \vee$; **б)** $\vee$ и $\to$ через $\neg, \wedge$; **в)** $\wedge$ и $\vee$ через $\neg, \to$; **г)** $\neg$ через $0, \to$; **д)** $\neg$ через $1, \oplus$; **е)** $\vee$ через $\to$; **ж)** $\neg, \vee, \wedge, \to, \sim$ через $\uparrow$; **з)** $\neg, \vee, \wedge, \to, \oplus$ через $|$; **и)** $\uparrow$ через $|$; **к)** $|$ через $\uparrow$.

386. Доказать полноту следующих систем функций сведением к заведомо полным системам:
а) $\{x_1 \uparrow x_2\}$; **б)** $\{x_1 \mid x_2\}$; **в)** $\{x_1 \to x_2, \overline{x_1 \oplus x_2 \oplus x_3}\}$; **г)** $\{(1011), (1100001100111100)\}$.

387. С помощью теоремы Поста проверить на полноту следующие системы функций:
а) x_1x_2, $x_1 \vee x_2$; **б)** $x_1 \to x_2$, $x_1 \to \overline{x}_2x_3$; **в)** $x_1\overline{x}_2$, $\overline{x}_1 \sim x_2x_3$; **г)** 0, 1, $x_1(x_2 \sim x_3) \vee \overline{x}_1(x_2 \oplus x_3)$; **д)** $\neg x$, (0010), (0101110011100011); **е)** 1, $x_1 \oplus x_2$, $(x_1 \to x_2) \uparrow (x_2 \sim x_3)$, $(x_3 \mid (x_1 \cdot x_2)) \to \overline{x}_3$; **ж)** $x_1 \to x_2$, $\overline{x}_1$;

з) x_1x_2, $x_1 \vee x_2$, $x_1 \to x_2$; **и)** $x_1 \sim x_2$, $\overline{x}_1$, $\overline{x}_1 \to \overline{x}_2$; **к)** $x_1 \to x_2$, 0, $x_1 \sim x_2$; **л)** $x_1 \oplus x_2$, $\overline{x}_1$; **м)** $x_1x_2 \vee x_1x_3 \vee x_2x_3$, 0, 1; **н)** $x_1x_2 \vee x_1x_3 \vee x_2x_3$, $\overline{x}_1$, $\overline{x}_1 \to x_2$;

388. Из полных систем примера 387 выделить все возможные базисы, то есть такие полные подсистемы, у которых ни одна собственная подсистема не является полной.

389. Доказать, что если система функций $\{f_1, f_2, \ldots, f_m\}$ полна, то и система функций $\{f_1^*, f_2^*, \ldots, f_m^*\}$ также полна.

390. Какие из следующих систем функций являются замкнутыми:
а) $P_2(1)$; **б)** $P_2(2)$; **в)** P_2; **г)** $P_0 \cap P_1$; **д)** $P_0 \cup P_1$; **е)** $P_0 \backslash P_1$.

391. Доказать, что пересечение функционально замкнутых классов является функционально замкнутым классом.

392. Доказать, что если множество $\mathcal{M}$ — функционально замкнутый класс, то множество $\mathcal{M}^*$, состоящее из функций, двойственных к функциям из $\mathcal{M}$, также является функционально замкнутым классом.

393. Доказать, что если $\mathcal{M} \neq \varnothing$, $\mathcal{M} \neq P_2$ и $[\mathcal{M}] = \mathcal{M}$, то $P_2 \backslash \mathcal{M}$ незамкнуто.

394. Обозначим через M^- множество монотонно убывающих булевых функций. Доказать, что множества M^- и $M \cup M^-$ незамкнуты.

395. Доказать, что для монотонности функции, отличной от константы, необходимо и достаточно, чтобы она представлялась в виде суперпозиции конъюнкций и дизъюнкций ($\Leftrightarrow f \in [\vee, \wedge]$).

396. Доказать, что $f \in M \Leftrightarrow f^* \in M$.

397. Найти $M \cap (P_2 \backslash P_0)$, $M \cap (P_2 \backslash P_1)$.

398. К какому наименьшему числу переменных можно свести немонотонную функцию с сохранением немонотонности, отождествляя её переменные?

399. Найти $P_2(2) \backslash (P_0 \cup P_1 \cup L \cup S \cup M)$.

400. Найти все функции, которые можно получить, отождествляя переменные, из следующих функций:
а) (10010110); **б)** (11111101); **в)** $x_1x_2 \vee x_2x_3 \vee x_1x_3$; **г)** $x_1x_2x_3 \oplus x_2x_3 \oplus x_3x_1 \oplus x_2 \oplus 1$.

§6. Релейно-контактные схемы и схемы из функциональных элементов

6.1. Задачи синтеза

Пример 11. Построить схему машины экзаменатора, в которой студенту предлагается вопрос и четыре варианта ответа на него, только один из ко-

торых правильный. В случае, когда ответ правильный, должно зажигаться табло “ответ верен”.

Решение. Закодируем номера ответов двухразрядными двоичными числами 00, 01, 10, 11. Студент и машина должны генерировать двухразрядные управляющие сигналы. Функция проводимости схемы задаётся таблицей

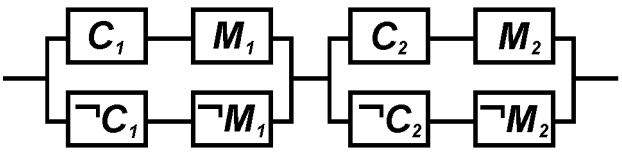
C_1	C_2	M_1	M_2	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Выпишем и упростим СДНФ функции f:

$$f \equiv \overline{C}_1\overline{C}_2\overline{M}_1\overline{M}_2 \vee \overline{C}_1C_2\overline{M}_1M_2 \vee C_1\overline{C}_2M_1\overline{M}_2 \vee C_1C_2M_1M_2 \equiv$$
$$\equiv \overline{C}_1\left(\overline{C}_2\overline{M}_2 \vee C_2M_2\right)\overline{M}_1 \vee C_1\left(\overline{C}_2\overline{M}_2 \vee C_2M_2\right)M_1 \equiv$$
$$\equiv \left(\overline{C}_2\overline{M}_2 \vee C_2M_2\right)\left(\overline{C}_1\overline{M}_1 \vee C_1M_1\right).$$

Схема имеет вид:

а)



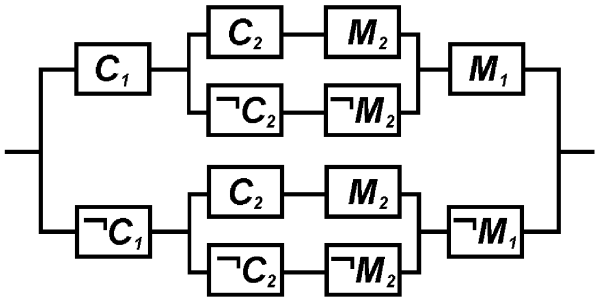
или

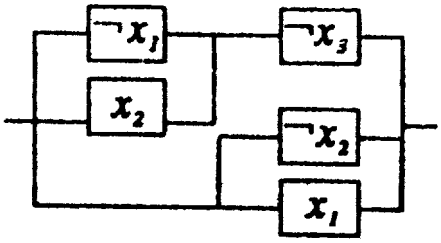
б)

Схема а) предпочтительней схемы б).

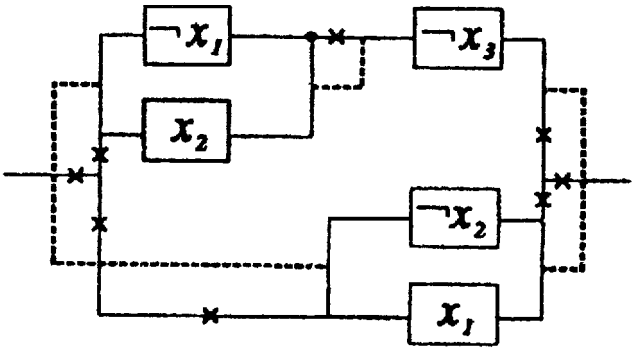
6.2. Анализ схем

Пример 12. Найти функцию проводимости схемы

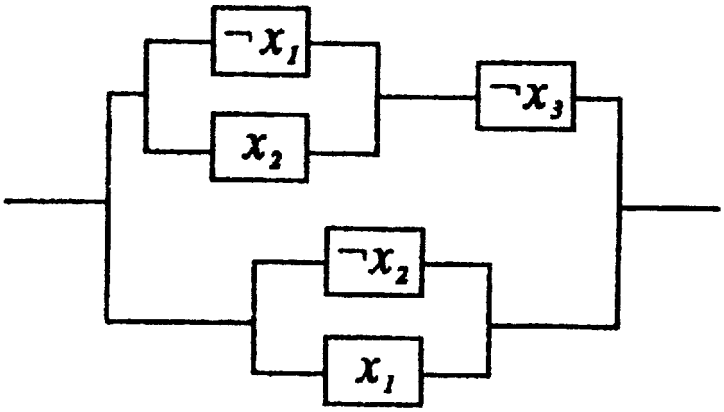




Решение. При решении задач такого типа следует помнить, что последовательное соединение реле соответствует конъюнкции, а параллельное — дизъюнкции. Полезным является умение преобразовать топологию схемы так, чтобы явно были видны последовательные и параллельные участки схемы. Преобразуем топологию схемы (добавленные участки обозначены пунктиром, удаляемые участки помечены “×”):



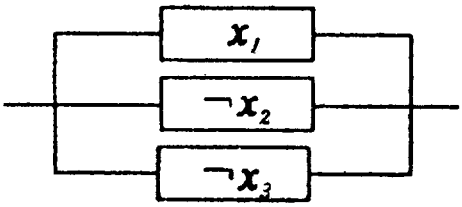
Получаем схему:



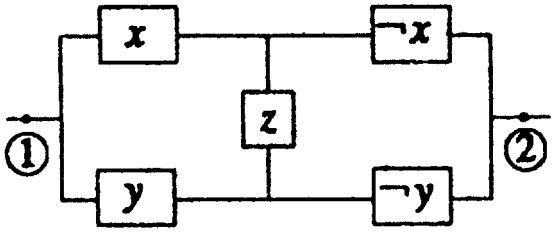
Её функция проводимости задаётся формулой

$$x_1 \vee \overline{x}_2 \vee (\overline{x}_1 \vee x_2)\overline{x}_3 \equiv x_1 \vee \overline{x}_2 \vee \overline{x}_1\,\overline{x}_3 \vee x_2\overline{x}_3 \equiv x_1 \vee \overline{x}_2 \vee \overline{x}_3.$$

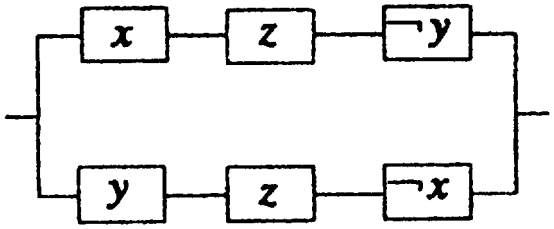
Значит, более простая схема имеет вид:



Замечание. Существуют схемы, в которых преобразование топологии не приводит к нужному результату (или такое преобразование трудно провести). Например, рассмотрим схему:



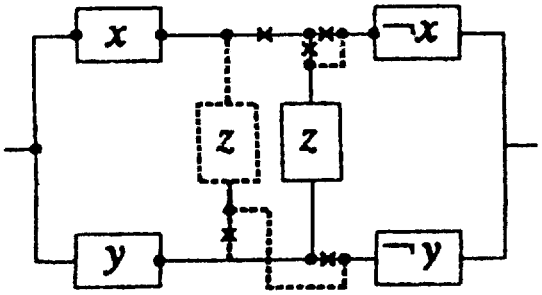
Анализ всевозможных путей прохождения по этой схеме от точки 1 до точки 2 показывает, что эквивалентная схема имеет следующий вид:



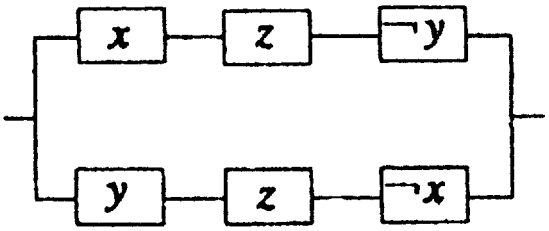
Функция проводимости исходной схемы задаётся формулой:

$$xz\overline{y} \vee \overline{x}yz.$$

Приведём теперь преобразование топологии схемы (здесь будут добавляться и удаляться не только проводники, но и реле):



Пересечение проводников, не отмеченное жирной точкой, означает их изоляцию друг от друга. Изобразим оставшееся на последней схеме.



Составить схемы, реализующие следующие функции:

401. $x \to y$; **402.** $x \sim y$; **403.** $x \vee \vee y$; **404.** $(x \to y)(y \to z)$; **405.** $(x \to y) \to \overline{x}(y \vee z)$;

406.

x	y	z	f_1	f_2	f_3
0	0	0	0	0	1
0	0	1	1	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0
1	1	1	0	0	1

407. Имеется одна лампа в лестничном пролёте двухэтажного дома. Построить схему так, чтобы на каждом этаже своим выключателем можно было гасить и зажигать лампу независимо от положения другого выключателя.

408. По установленному сигналу каждый игрок замыкает или размыкает выключатель, находящийся под его управлением. Если оба делают одно и то же, то выигрывает A, в противном случае — B. Построить схему так, чтобы в случае выигрыша A зажигалась лампочка.

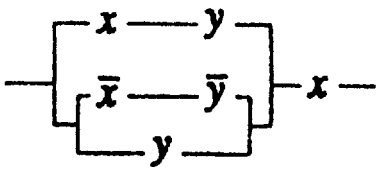
409. Комитет из 5 человек принимает решения большинством голосов. Председатель пользуется правом "вето". Построить схему так, чтобы голосование происходило нажатием кнопок и в случае принятия решения загоралась лампочка.

410. Построить схему, управляющую спуском лифта со второго этажа на первый. Условия, определяющие работу лифта, следующие:

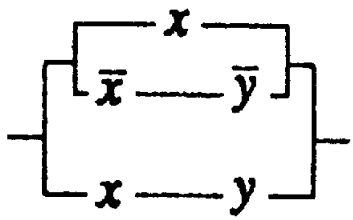
— дверь лифта на первом этаже закрыта,
— дверь лифта на втором этаже закрыта,
— пассажир находится в кабине лифта,
— кнопка вызова на первом этаже нажата,
— кнопка спуска на первый этаж в кабине нажата.

Найти функции проводимости следующих схем, если возможно, упростить схемы:

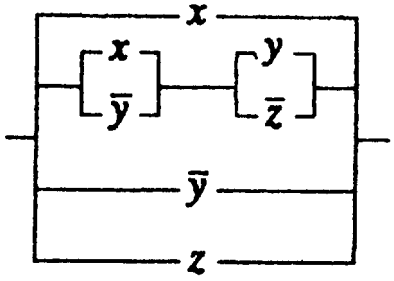
411.



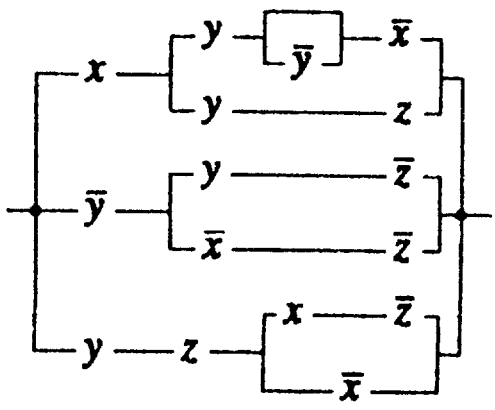
412.



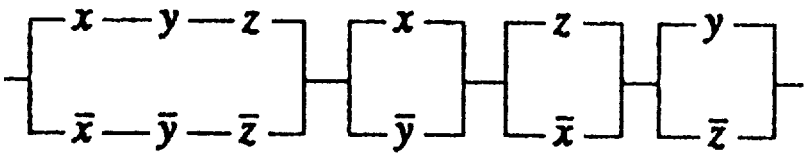
413.



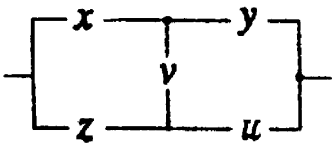
414.



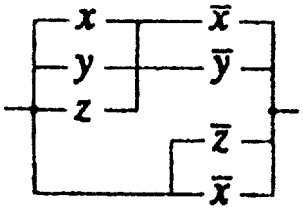
415.



416.



417.



§7. Предикаты, кванторы

При решении примеров на доказательство равносильности формул алгебры предикатов следует обращать внимание на следующее.

1. Области определения предикатов, стоящих слева и справа от знака $\equiv$, должны совпадать.

2. Связанная квантором переменная может обозначаться любой буквой, то есть

$$\forall x P(x) \equiv \forall y P(y) \equiv \forall t P(t) \equiv \dots$$

Какие из следующих предложений являются предикатами?

418. x делится на 3 ($x \in \mathbb{N}$); **419.** x делится на 5; **420.** $y = x^2$, $x \in \mathbb{R}$; **421.** $x^2 + x + 1$, $x \in \mathbb{R}$; **422.** $x^2 + y^2 = 0$, $x, y \in \mathbb{R}$; **423.** $x^2 + y^2 \geqslant 0$, $x, y \in \mathbb{R}$; **424.** $x^2 + y^2 = z$, $x, y, z \in \mathbb{R}$; **425.** $x < y$, $x, y \in \mathbb{R}$; **426.** Для всякого $x \in \mathbb{R}$ найдётся $y \in \mathbb{R}$ такой, что $x = y + 1$. **427.** $x^2 + y^2 < -2$, $x, y \in \mathbb{R}$.

428. Какие из предикатов в примерах 418 — 427 тождественно истинны, тождественно ложны, нетривиально выполнимы?

Выделить свободные переменные следующих предикатов:

429. $\forall x(x - y \equiv x + (-y),\ x, y \in \mathbb{R})$; **430.** $(x < y,\ x, y \in \mathbb{R}) \to$ $\to \exists z((x < z) \wedge (z < y),\ z \in \mathbb{R}$; **431.** $\forall y((y \in \mathbb{R},\ y > 0) \to \exists z(x = yz,\ x, z \in \mathbb{R}))$; **432.** $\forall x(\exists y P(x, y) \to Q(x, y, z))$; **433.** $\exists u \forall v \Phi(u, v) \to \exists t \Phi(t, u)$.

434. Из предикатов примеров 418 — 427 образовать с помощью кванторов высказывания, найти их значения истинности.

Доказать следующие равносильности:

435. $\overline{\forall x P(x, y)} \equiv \exists x \overline{P(x, y)}$; **436.** $\overline{\exists x P(x, y)} \equiv \forall x \overline{P(x, y)}$; **437.** $\forall x \forall y P(x, y, z) \equiv \forall y \forall x P(x, y, z)$; **438.** $\exists x \exists y P(x, y, z) \equiv \exists y \exists x P(x, y, z)$;

439. $\forall x(P(x,y)\wedge Q(x,y)) \equiv \forall x P(x,y)\wedge\forall x Q(x,y)$; **440.** $\exists x(P(x,y)\vee Q(x,y)) \equiv$ $\equiv \exists x P(x,y) \vee \exists x Q(x,y)$; **441.** $\forall x(P(x,z) \vee Q(y,z)) \equiv \forall x P(x,z) \vee Q(y,z)$; **442.** $\exists x(P(x,z) \wedge Q(y,z)) \equiv \exists x P(x,z) \wedge Q(y,z)$; **443.** $\exists x \forall y P(x,y,z) \to$ $\to \forall y \exists x P(x,y,z) \equiv 1$.

Ввести предикаты и с помощью кванторов записать следующие определения, с помощью законов де Моргана получить их отрицания:

444. определение предела числовой последовательности;

445. определение предела функции в точке;

446. определение непрерывности функции в точке;

447. определение непрерывной на интервале функции;

448. определение равномерно непрерывной на интервале функции.

449. Почему из равномерной непрерывности функции на (a,b) следует непрерывность функции на (a,b)?

450. Доказать, что существуют предикаты Φ, Q и P такие, что

а) $\forall x(\Phi(x) \vee Q(x)) \neq \forall x\Phi(x) \vee \forall x Q(x)$;
б) $\exists x(\Phi(x) \wedge Q(x)) \neq \exists x\Phi(x) \wedge \exists x Q(x)$;
в) $\forall y \exists x P(x,y) \to \exists x \forall y P(x,y) \neq 1$.

451. Какие из следующих формул тождественно истинны?

а) $\forall x(\Phi(x) \to P(x)) \to (\forall x\Phi(x) \to \forall x P(x)$;
б) $\forall x(\Phi(x) \to P(x)) \to (\exists x\Phi(x) \to \exists x P(x)$;
в) $\exists x(\Phi(x) \to P(x)) \to (\forall x\Phi(x) \to \forall x P(x)$;
г) $\exists x(\Phi(x) \to P(x)) \to (\exists x\Phi(x) \to \exists x P(x)$;
д) $\forall x(\Phi(x) \to P(x)) \sim (\exists x\Phi(x) \to \forall x P(x)$.

§8. Машина Тьюринга

В этом разделе содержатся задачи двух типов:

— по заданной машине Тьюринга найти результат её применения к заданному слову u, то есть найти $T(u)$;

— построить машину, решающую данный класс задач.

При решении задач второго типа рекомендуется до составления программы машины, то есть до заполнения таблицы, задающей программу, тщательно продумать алгоритм, который должен быть реализован программой. В конце решения (когда программа составлена) не забудьте применить её к тестовому примеру.

По заданной машине T с внешним алфавитом $A = \{|, \wedge\}$ и слову u найти слово $T(u)$:

452.

	q_1	q_2
$\|$	$\wedge q_2+1$	$\| q_2-1$
$\wedge$	$\| q_0\,0$	$\wedge q_1+1$

$u_1 = \|\|\|\|$
$u_2 = \| \wedge\wedge \|$

453.

	q_1	q_2	q_3
$\|$	$\| q_3+1$	$\| q_2\,0$	$\| q_1+1$
$\wedge$	$\| q_2+1$	$\| q_3+1$	$\| q_0\,0$

$u_1 = \|\|\|$
$u_2 = \| \wedge\wedge \|$
$u_3 = \|\| \wedge \wedge \wedge \|$

Выяснить, применима ли машина T с внешним алфавитом $\{|, \wedge\}$ к слову u, и в случае применимости найти результат:

454.

	q_1	q_2
$\|$	$\wedge q_1+1$	$\wedge q_2-1$
$\wedge$	$\wedge q_2-1$	$\| q_0+1$

$u_1 = \|\|\|$
$u_2 = \|\| \wedge \|$

455.

	q_1	q_2	q_3
$\|$	$\wedge q_1+1$	$\| q_1-1$	$\| q_2+1$
$\wedge$	$\wedge q_2+1$	$\wedge q_3+1$	$\wedge q_0\,0$

$u_1 = \|\| \wedge \|$
$u_2 = \| \wedge \|\|\|\|$

456. Построить машину K_2 над алфавитом $\{|\}$.

457. Построить машину K_1 над алфавитом $\{\alpha, \beta\}$.

458. Какую функцию натурального аргумента вычисляет машина, заданная программой:

	q_1	q_2
$\|$	$\| q_2+1$	$\| q_2+1$
$\wedge$	$\| q_0\,0$	$\| q_0\,0$

Упростить эту машину.

459. Построить машину, распознающую чётность натурального числа.

460. Построить машину R_m, вычисляющую остаток от деления натурального числа на m.

461. Построить машины Тьюринга, вычисляющие следующие функции, заданные на $\mathbb{N} \times \mathbb{N}$:

а) $x + y$; **б)** $x + 2y$; **в)** $x \cdot y$; **г)** $x^2 + 3y$.

462. Построить машины Тьюринга, вычисляющие следующие функции, определённые на $\mathbb{N}$:

а) $3x$; **б)** x^2; **в)** $f(x) = \begin{cases} x+1 & \text{при } x = 2n, \\ 2x & \text{при } x = 2n+1. \end{cases}$

463. Построить машину над алфавитом $\{|\}$, применимую к любому слову чётной длины и не применимую к словам нечётной длины.

Построить в алфавите $\{0,1\}$ машину T, работающую по правилу:

464. $T(1^n) = 1^n01^n$, $n \in \mathbb{N}$, $a^n \stackrel{\text{def}}{=} \underbrace{aa\ldots a}_{n}$;

465. $T(0^n1^n) = (01)^n$, $n \in \mathbb{N}$;

466. $T(1^n) = 1^n01^{2n}01^{3n}$, $n \in \mathbb{N}$;

467. $T(1^n01^m) = 1^m01^n$, $n, m \in \mathbb{N}$;

468. $T(1^n0^m) = \begin{cases} 1^{2n} & \text{при } n > m, \\ (01)^n & \text{при } n = m, \\ 0^m & \text{при } n < m. \end{cases}$

469. Какую функцию натурального аргумента вычисляет машина T?

а)

	q_1	q_2	q_3	q_4	q_5
$\|$	$\| q_1+1$	$0q_3+1$	$0q_3+1$	$\| q_5-1$	$\| q_5-1$
$\wedge$	$\wedge q_2+1$	$\wedge q_1-1$	$\wedge q_4-1$	$\wedge q_4-1$	$\wedge q_0+1$

б)

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9
$\|$	$\wedge q_2{+}1$	$\|q_4{+}1$	$\|q_3{-}1$	$\|q_4{+}1$	$\|q_6{+}1$	$\|q_6{+}1$	$\wedge q_8{-}1$	$\|q_8{-}1$	$\|q_9{+}1$
$\wedge$	$\wedge q_2{+}1$	$\wedge q_3{+}1$	$\|q_0\,0$	$\wedge q_5{+}1$	$\wedge q_3{-}1$	$\wedge q_7{-}1$		$\wedge q_9{-}1$	$\wedge q_1{+}1$

470. Какие одноместные функции натурального аргумента в алфавите $\{|, \wedge\}$ могут вычислять машины, программы которых содержат только команды q_0 и q_1?

По словесному описанию машин $T_3, T_4, \ldots$ построить их программы с внешним алфавитом $\{0, 1, \wedge\}$:

471. T_3 — начиная с последней единицы массива из единиц, "сдвигает" его на одну ячейку влево и останавливается на первой единице;

472. T_4 — при заданном $l \geqslant 1$ СЗУ машины, начав с произвольной ячейки, заполненной единицей, движется вправо, не меняя содержимого ячеек, до тех пор пока не пройдёт массив из $l+1$ нуля; СЗУ останавливается на следующей ячейке, поместив туда единицу;

473. T_5 — при заданном $l \geqslant 1$ СЗУ, начав с произвольной ячейки и двигаясь вправо, проставляет подряд l единиц и останавливается на последней из них;

474. T_6 — машина начинает работу с крайней слева непустой ячейки произвольного слова, при заданном $l \geqslant 1$ отыскивает в слове первый слева

массив из $l+1$ нуля и останавливается на последнем из них (содержимое ячеек не меняется);

475. T_7 — начав работу с самой левой непустой ячейки, машина отыскивает единицу, примыкающую слева к первому слева массиву из трёх нулей, "окаймлённому" единицами, СЗУ останавливается на найденной единице (содержимое ячеек не меняется);

476. T_8 — в исходной ячейке печатает нуль, СЗУ сдвигается на одну ячейку влево и машина останавливается.

477. T_9 — СЗУ машины сдвигается на две ячейки вправо от начальной, машина останавливается в состоянии q_0, если эта ячейка содержит нуль, в состоянии q_0', если эта ячейка содержит единицу.

478. T_{10} — СЗУ передвигается на одну ячейку влево и машина останавливается;

479. T_{11} — отправляясь от начальной ячейки, находит первую единицу и останавливается на следующей за ней ячейке.

480. Найдите композиции машин: $T_4 \circ T_3$, $T_6 \circ T_7$, $T_{11} \circ T_{10} \circ T_5$. (Машины $T_3, T_4, \ldots$ см. в примерах 471 — 479.)

§9. Вычислимые функции

Проверка вычислимости, перечислимости и разрешимости множества состоит в предъявлении вычислимой функции, т.е. алгоритма. Алгоритм может быть представлен в виде блок-схемы или написан на одном из распространенных языков (C, Pascal, etc.).

Проверить:

481. Разрешимость множества простых чисел

482. Перечислимость графика вычислимой функции

483. Разрешимость любого конечного множества

484. Разрешимость $2\mathbb{N}$

485. Привести пример перечислимого неразрешимого множества.

486. Привести пример неразрешимого множества.

487. Разрешимость множества всех рациональных чисел, меньших 3.

488. Арифметичность предиката "$x = n$- тое по порядку простое число".

Доказать арифметичность предикатов в сигнатуре $(=,)$ на множестве натуральных чисел :

489. $x < y$,

490. $x = 0$,

491. $x = 1$,

492. x делится на y без остатка,

493. x - простое число.

494. изобразить соответствующее предикату $x < y$ арифметическое множество в $\mathbb{N}^2$

Список литературы

[1] Н. К. Верещагин, А. Шень. *Лекции по математической логике и теории алгоритмов. Часть 1. Начала теории множеств.* М.: МЦНМО, 1999. 128 с.

[2] Н. К. Верещагин, А. Шень. *Лекции по математической логике и теории алгоритмов. Часть 2. Языки и исчисления.* М.: МЦНМО, 2000. 288 с.

[3] Н. К. Верещагин, А. Шень. *Лекции по математической логике и теории алгоритмов. Часть 3. Вычислимые функции.* М.: МЦНМО, 1999. 176 с.

[4] Я. М. Ерусалимский, *Дискретная математика: теория, задачи, приложения.* 4-е издание - М.: Вузовская книга, 2001. – 280 с.

[5] Ю. И. Манин. *Доказуемое и недоказуемое.* М.: Советское радио, 1979. 168 с.

[6] Ю. И. Манин, *Вычислимое и невычислимое.* М.: Советское радио, 1980. 128 с.