С. В. ЯБЛОНСКИЙ

ВВЕДЕНИЕ В ДИСКРЕТНУЮ МАТЕМАТИКУ

С. В. ЯБЛОНСКИЙ

ВВЕДЕНИЕ В ДИСКРЕТНУЮ МАТЕМАТИКУ

ИЗДАНИЕ ВТОРОЕ, ПЕРЕРАБОТАННОЕ И ДОПОЛНЕННОЕ

Допущено Министерством высшего и среднего специального образования СССР в качестве учебного пособия для студентов вузов, обучающихся по специальности «Прикладная математика»

МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1986

ББК 22.18
Я14
УДК 519.6(075.8)

Яблонский С. В. **Введение в дискретную математику:** Учеб. пособие для вузов.— 2-е изд., перераб. и доп.— М.: Наука. Гл. ред. физ.-мат. лит.— 384 с.

Книга является введением в дискретную математику — раздел прикладной математики, бурно развивающийся в последние годы и являющийся базой для математической кибернетики. Она написана на основе курса лекций, читавшегося автором в течение ряда лет на факультете вычислительной математики и кибернетики Московского государственного университета.

Предназначается студентам факультетов прикладной математики, аспирантам, а также инженерам и специалистам, работающим в области прикладной математики.

Первое издание вышло в 1979 г.
Табл. 54. Ил. 173. Библиогр. 50 назв.

Сергей Всеволодович Яблонский

ВВЕДЕНИЕ В ДИСКРЕТНУЮ МАТЕМАТИКУ

Редактор *В. М. Храпченко*
Художественный редактор *Т. Н. Кольченко*
Технический редактор *Л. В. Лихачева*
Корректор *М. Л. Медведская*

ИБ № 12366

Сдано в набор 18.02.86. Подписано к печати 11.09.86. Формат 84×108¹/₃₂. Бумага тип. № 2. Гарнитура обыкновенная. Печать высокая. Усл. печ. л. 20,16. Усл. кр.-отт. 20,16. Уч.-изд. л. 19,76. Тираж 22 000 экз. Заказ № 61. Цена 95 коп.

Ордена Трудового Красного Знамени издательство «Наука»
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15

4-я типография издательства «Наука»
630077 г. Новосибирск, 77, Станиславского, 25

Л $\frac{1702070000-157}{053(02)-86}$ 66-86

ОГЛАВЛЕНИЕ

ЧАСТЬ V

НЕКОТОРЫЕ ПРИЛОЖЕНИЯ К КИБЕРНЕТИКЕ

ПРЕДИСЛОВИЕ

Дискретная математика — часть математики, которая зародилась в глубокой древности. Как говорит само заглавие, главной ее спецификой является дискретность, т. е. антипод непрерывности. В широком смысле дискретная математика включает в себя и такие сложившиеся разделы математики, как теория чисел, алгебра, математическая логика и ряд разделов, которые наиболее интенсивно стали развиваться в середине этого века в связи с внедрением ЭВМ. Научно-технический прогресс поставил проблему изучения сложных управляющих систем. В узком смысле дискретная математика ограничивается только этими новыми разделами. Именно так это понимается и в данной книге. К упомянутым новым разделам мы относим: теорию функциональных систем; теорию графов и сетей; теорию кодирования; комбинаторный анализ; целочисленное программирование и т. п.

Дискретная математика сегодня является не только фундаментом математической кибернетики, но и важным звеном математического образования. Книга содержит материал, соответствующий двум типам программ курса «Дискретная математика»: стандартной программе для факультетов прикладной математики и кибернетики большинства университетов и программе соответствующего курса, читаемого в МГУ. Эти программы не ставят целью дать большое количество материала фактического и имеющего спрос в данное время. Чрезмерная детализация и привязывание программы к специальным фактам опасны тем, что лет через 10—15 (а это как раз время активной деятельности обучаемых сейчас студентов) появятся новые факты, а старые частично утратят свою значимость. Ввиду этого главная задача курса — это обучение методам и мышлению, характерным для дискретной математики. Материал, вошедший в эту книгу, знакомит читателя с узловыми задачами из нескольких раз-

делов дискретной математики и ее приложений. Он подобран таким образом, чтобы сократить число необходимых понятий до минимума и, с другой стороны, дать небольшое количество (10—15) серьезных теорем с непохожими доказательствами, а также познакомить с применениями понятия алгоритма, владение которым особенно важно для специалистов в области прикладной математики.

В основу данной книги положен курс, который впервые был прочитан автором на механико-математическом факультете МГУ в 1964 г.

Книга состоит из пяти частей.

I часть. Функциональные системы с операциями.

II часть. Комбинаторный анализ.

III часть. Графы и сети.

IV часть. Теория кодирования.

V часть. Некоторые приложения к кибернетике.

Таким образом, содержание книги охватывает почти все основные разделы дискретной математики. При ее написании автору пришлось усовершенствовать целый ряд доказательств, в ряде случаев дать новые доказательства, которые публикуются здесь впервые.

Изложение многих вопросов из перечисленных разделов не всегда ведется на абстрактной основе: здесь широко используется геометрический язык и содержательные интерпретации. Это позволяет сочетать в построениях наглядность и известную строгость. Надо иметь в виду, что в математических статьях существуют две крайности: пренебрежение строгостью изложения и доведение строгости до абсурда. Обе указанные тенденции одинаково опасны. Строгость изложения должна соответствовать рассматриваемой задаче (и уровню аудитории), подобно тому как в приближенных вычислениях число значащих цифр получаемых результатов должно соответствовать точности исходных данных. Слишком «большой» запас строгости в этом смысле подобен вычислениям со значительным числом дополнительных разрядов.

Данная книга рассчитана на студентов, аспирантов и научных сотрудников.

При пользовании материалом, изложенным в книге, надо иметь в виду, что содержание частей I, III, IV соответствует программе курса «Введение в дискретную математику» для студентов МГУ. Для изучения дискретной

математики по типовой программе факультетов прикладной математики университетов части I, III, IV следует брать в сокращенном варианте, но к ним добавляется часть V. Эти сокращения состоят примерно в следующем: исключается гл. 2 (часть I), кроме § 1; в гл. 3 (часть I) опускается материал до канонических уравнений; в гл. 4 (часть I) исключается последний параграф; в гл. 2 (часть III) опускается теорема о числе сетей. Некоторый вспомогательный материал можно найти в части II.

Второе издание книги отличается от первого наличием дополнительной части по комбинаторному анализу, некоторой переработкой остальных частей и устранением всех замеченных погрешностей. В этой работе автору помогли замечания многих математиков, которым автор весьма признателен. Особую благодарность автор выражает редакторам первого и второго изданий Е. П. Липатову и В. М. Храпченко, проделавшим большую работу по улучшению изложения материала.

С. В. Яблонский

ЧАСТЬ I

ФУНКЦИОНАЛЬНЫЕ СИСТЕМЫ С ОПЕРАЦИЯМИ

Теория функциональных систем занимается изучением функций, описывающих работу дискретных преобразователей. Здесь рассматриваются важнейшие классы функций: булевы функции, функции k-значной логики, автоматные (о.-д.) функции и вычислимые функции. С каждым из этих классов естественным образом связываются операции, позволяющие из одних функций данного класса строить другие функции этого же класса. Такими операциями являются операция суперпозиции, операция обратной связи, операция примитивной рекурсии и μ-операция. В результате этого мы приходим к функциональным системам с операциями — некоторым классам алгебр. Данные объекты в книге расположены так, что каждый последующий объект является «расширением» предыдущего. Это позволяет переносить результаты с простых систем на более сложные. При этом обращается внимание на аналогии и на существенные различия.

Роль теории функциональных систем в дискретной математике можно сравнить с ролью математического анализа в непрерывной математике.

Глава 1

АЛГЕБРА ЛОГИКИ

§ 1. Функции алгебры логики

Пусть $U = \{u_1, u_2, \ldots, u_m, \ldots\}$ — исходный алфавит переменных (аргументов). Будем рассматривать функции $f(u_{i_1}, u_{i_2}, \ldots, u_{i_n})$ $(u_{i_\nu} \neq u_{i_\mu}$ при $\nu \neq \mu)$, аргументы которых определены на множестве $E_2 = \{0, 1\}$, и такие, что $f(\alpha_1, \alpha_2, \ldots, \alpha_n) \in E_2$, когда $\alpha_i \in E_2$ $(i = 1, 2, \ldots, n)$. Эти функции будем называть *функциями алгебры логики* или *булевыми функциями*. Чтобы избежать

сложных обозначений для индексов переменных, мы будем употреблять в качестве метаобозначений (обозначений для произвольных символов алфавита U) символы x, y, z, ..., а также эти символы с индексами. Таким образом, запись $f(x_1, x_2, \ldots, x_n)$ понимается как запись функции, зависящей от произвольных фиксированных аргументов $u_{i_1}, u_{i_2}, \ldots, u_{i_n}$, где $u_{i_\nu} \neq u_{i_\mu}$ при $\nu \neq \mu$.

Таблица 1

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

Из определения функции $f(x_1, x_2, \ldots, x_n)$ следует, что для ее задания достаточно указать, какое значение функции соответствует каждому из наборов значений аргументов, т. е. выписать таблицу (см. табл. 1). Легко видеть, что n переменных принимают 2^n различных значений. Для удобства мы употребляем стандартное расположение наборов: если набор рассматривать как запись числа в двоичном исчислении, то расположение наборов соответствует естественному порядку следования чисел $0, 1, \ldots, 2^n - 1$. Далее мы видим, что каждая функция $f(x_1, x_2, \ldots, x_n)$ определяет отображение

$$\underbrace{E_2 \times E_2 \times \ldots \times E_2}_{n \text{ раз}} \to E_2.$$

Поэтому естественно интерпретировать символ f как символ, обозначающий это отображение, а $x_1, x_2, \ldots, x_n$ — как названия столбцов. В этом случае функции

$$f(x_1, x_2, \ldots, x_n), \quad f(y_1, y_2, \ldots, y_n)$$

будут задавать одно и то же отображение, и их таблицы будут отличаться только, быть может, названиями столбцов. Обозначим через P_2 систему всех функций алгебры логики над алфавитом U, содержащую также константы 0 и 1.

Если зафиксировать n переменных $x_1, x_2, \ldots, x_n$, то различные таблицы будут отличаться лишь значениями

правого столбца. Поэтому справедливо следующее утверждение.

Теорема 1. *Число* $p_2(n)$ *всех функций из* P_2, *зависящих от* n *переменных* $x_1, x_2, \ldots, x_n$, *равно* 2^{2^n}.

Здесь следует обратить внимание на два обстоятельства.

1. Число функций алгебры логики, зависящих от заданных n аргументов, конечно. Поэтому, если нужно выяснить, обладают ли функции из этого конечного множества каким-либо свойством, достаточно осуществить просмотр (или, как говорят, «перебор») функций из данного множества. Однако числа $p_2(n)$ с ростом n быстро растут:

$$p_2(1)=4,\ p_2(2)=16,\ p_2(3)=256,\ p_2(4)=65536,\ \ldots$$

Следовательно, уже при сравнительно небольших значениях n ($n \geqslant 6$) перебор становится практически невозможным даже с использованием вычислительной техники.

2. С ростом числа аргументов таблица, задающая функцию, сильно усложняется. Так, например, уже при не очень большом числе аргументов, скажем при $n = 10$, таблица становится громоздкой (имеет 1024 строки), а при $n = 20$ — практически необозримой.

Введенное выше понятие функции несовершенно, поскольку оно не позволяет рассматривать функции от меньшего числа аргументов как функции от большего числа аргументов. Для устранения этого недостатка введем следующее определение.

Определение. Функция $f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ из P_2 *зависит существенным образом* от аргумента x_i, если существуют такие значения $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$ переменных $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, что

$$f(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n) \neq f(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n).$$

В этом случае переменная x_i называется *существенной*. Если x_i не является существенной переменной, то она называется *несущественной* или *фиктивной*.

Пусть для функции $f(x_1, \ldots, x_n)$ переменая x_i является фиктивной. Возьмем таблицу для функции $f(x_1, \ldots, x_n)$ и по ней построим новую таблицу путем вы-

черкивания всех строк вида $(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n)$ и вычеркивания столбца для аргумента x_i. Полученная таблица будет определять некоторую функцию $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Будем говорить, что функция $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ получена из $f(x_1, \ldots, x_n)$ *путем удаления фиктивной переменной* x_i, а также, что функция $f(x_1, \ldots, x_n)$ получается из $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ *путем введения фиктивной переменной* x_i.

Определение. Функции f_1 и f_2 называются *равными*, если функцию f_2 можно получить из f_1 путем добавления и изъятия фиктивных аргументов.

В дальнейшем всюду функции рассматриваются с точностью до фиктивных переменных, т. е. мы считаем, что если задана функция f_1, то задана и любая равная ей функция f_2. Это накладывает некоторые естественные ограничения на классы функций, которые будут здесь рассматриваться. В частности, класс функций, обладающих определенными свойствами, будет рассматриваться только в том случае, если эти свойства инвариантны относительно операций введения и удаления фиктивных переменных.

Существуют два типа функций, которые не имеют существенных переменных: функции первого типа тождественно равны 0, а второго — 1. Ввиду этого целесообразно включить в наши рассмотрения константы 0 и 1 (рассматривая их как функции от пустого множества переменных).

Для иллюстрации сказанного рассмотрим два класса симметрических функций.

Определение. Булева функция $f(x_1, \ldots, x_n)$ называется *симметрической относительно переменных* $x_1, \ldots, x_k$ $(2 \leqslant k \leqslant n)$, если для любой подстановки $\begin{pmatrix} 1 \ldots k \\ j_1 \ldots j_k \end{pmatrix}$ имеет место равенство

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) = f(x_{j_1}, \ldots, x_{j_k}, x_{k+1}, \ldots, x_n).$$

Аналогично вводится понятие функции, симметрической относительно произвольных переменных $x_{i_1}, \ldots, x_{i_k}$.

Очевидно, что функции, тождественно равные константам 0 и 1, являются симметрическими относительно любой совокупности своих переменных.

Пример 1. а) Класс функций, симметрических относительно всех своих переменных. Это свойство не яв-

ляется инвариантным относительно операции введения несущественной переменной, так как любая функция, отличная от константы и обладающая этим свойством, существенно зависит от всех своих переменных (зависимость хотя бы от одной переменной следует из того, что функция принимает оба значения 0 и 1, а зависимость от всех переменных — из симметрии). Такие классы дальше рассматриваться не будут.

б) Класс функций, симметрических относительно всех своих существенных переменных. Это свойство, очевидно, является инвариантным относительно операций введения и удаления несущественных переменных. Такой класс может рассматриваться.

Замечание. Если дана конечная система функций из P_2: $\{f_1, \ldots, f_s\}$, $s \geqslant 1$, то можно считать, что все эти функции зависят от одних и тех же переменных $x_1, \ldots, x_n$, т. е. имеют вид

$$f_1(x_1, \ldots, x_n), \ldots, f_s(x_1, \ldots, x_n).$$

С другой стороны, если дана функция, отличная от константы, то путем отождествления переменных из нее можно получить равную ей функцию, все переменные которой являются существенными.

В заключение этого параграфа рассмотрим примеры функций алгебры логики. Данные функции часто употребляются в математической логике и кибернетике и играют такую же роль, как, например, x^n или $\sin x$ в анализе, поэтому их можно считать «элементарными»:

1) $f_1(x) = 0$ — *константа* 0;

2) $f_2(x) = 1$ — *константа* 1;

3) $f_3(x) = x$ — *тождественная функция*;

4) $f_4(x) = \bar{x}$ — *отрицание* x ($\bar{x}$ читается «не x»);

5) $f_5(x_1, x_2) = (x_1 \& x_2)$ — *конъюнкция* x_1 *и* x_2 (читается «x_1 и x_2»). Вместо знака & употребляется знак $\cdot$ или вообще знак опускается, т. е. пишут (x_1x_2). Эту функцию часто называют *логическим умножением*;

6) $f_6(x_1, x_2) = (x_1 \vee x_2)$ — *дизъюнкция* x_1 *и* x_2 (читается «x_1 или x_2»). Эту функцию часто называют *логическим сложением*;

7) $f_7(x_1, x_2) = (x_1 \to x_2)$ — *импликация* x_1 *и* x_2 (читается «из x_1 следует x_2»). Эту функцию часто называют *логическим следованием*;

8) $f_8(x_1, x_2) = (x_1 + x_2)$ — *сложение* x_1 *и* x_2 *по* $\bmod 2$;

9) $f_9(x_1, x_2) = (x_1/x_2)$ — *функция Шеффера.*

Таблица 2

x	0	1	x	$\bar{x}$
0	0	1	0	1
1	0	1	1	0

Таблица 3

x_1	x_2	$(x_1 \& x_2)$	$(x_1 \vee x_2)$	$(x_1 \to x_2)$	$(x_1 + x_2)$	(x_1 / x_2)
0	0	0	0	1	0	1
0	1	0	1	1	1	1
1	0	0	1	0	1	1
1	1	1	1	1	0	0

В таблицах 2, 3 даются значения этих функций. Заметим, что

$$(x_1 \& x_2) = \min(x_1, x_2) = (x_1 \cdot x_2), \quad (x_1 \vee x_2) = \max(x_1, x_2).$$

§ 2. Формулы. Реализация функций формулами

Как и в элементарной алгебре, исходя из «элементарных» функций, можно строить формулы. Ниже приводится индуктивное определение формул.

Определение. Пусть $\mathfrak{P}$ — некоторое (не обязательно конечное) подмножество функций из P_2.

а) Базис индукции. Каждая функция $f(x_1, \ldots, x_m)$ из $\mathfrak{P}$ называется *формулой над* $\mathfrak{P}$.

б) Индуктивный переход. Пусть $f_0(x_1, \ldots, x_m)$ — функция из $\mathfrak{P}$ и $A_1, \ldots, A_m$ — выражения, являющиеся либо формулами над $\mathfrak{P}$, либо символами переменных из U. Тогда выражение $f_0(A_1, \ldots, A_m)$ называется *формулой над* $\mathfrak{P}$.

Пример 1. Пусть $\mathfrak{P}$ — множество «элементарных» функций. Следующие выражения являются формулами над $\mathfrak{P}$:

1) $\{[(x_1 x_2) + x_1] + x_2\}$;
2) $[\bar{x}_1 (x_2 + x_3)]$;
3) $\overline{\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}}$.

В дальнейшем будем обозначать формулы заглавными готическими буквами с квадратными скобками, в которых перечисляются функции, необходимые для их построения. Так

$$\mathfrak{A}[f_1, \ldots, f_s]$$

означает, что формула $\mathfrak{A}$ построена из $f_1, \ldots, f_s$. В тех случаях, когда нужно обратить внимание на множество тех переменных, которые участвуют в построении формулы, пишут

$$\mathfrak{A}(x_1, \ldots, x_n).$$

Пусть $\mathfrak{A}$ — произвольная формула над $\mathfrak{P}$, тогда формулы, которые использовались для ее построения, будем называть *подформулами* формулы $\mathfrak{A}$.

Пусть формула $\mathfrak{A}$ является формулой над множеством $\mathfrak{P} = \{f_1(x_1, \ldots, x_n), \ldots, f_s(x_1, \ldots, x_n)\}$, т. е. $\mathfrak{A} = \mathfrak{A}[f_1, \ldots, f_s]$. Возьмем множество функций

$$\mathfrak{Q} = \{g_1(x_1, \ldots, x_n), \ldots, g_s(x_1, \ldots, x_n)\},$$

где g_i имеет те же переменные, что и f_i $(i = 1, \ldots, s)$.

Определение. Рассмотрим формулу $\mathfrak{B} = \mathfrak{B}[g_1, \ldots, g_s]$, которая получается из $\mathfrak{A}$ путем замены $\begin{pmatrix} f_1 & \ldots & f_s \\ g_1 & \ldots & g_s \end{pmatrix}$. Говорят, что формула $\mathfrak{B}$ имеет *то же строение*, что и формула $\mathfrak{A}$.

Пример 2.

1) $\mathfrak{P} = \{\bar{x}_1, (x_1 \& x_2)\}$, $\mathfrak{A} = [x_1 \& \overline{(x_2 \& x_3)}]$;
2) $\mathfrak{Q} = \{\bar{x}_1, (x_1 \to x_2)\}$, $\mathfrak{B} = [x_1 \to (x_2 \to x_3)]$.

Очевидно, что $\mathfrak{A}$ и $\mathfrak{B}$ имеют одинаковое строение.

В дальнейшем строение формулы обозначается через C (с индексами или без них), и формула $\mathfrak{A}$ однозначно определяется строением C и упорядоченной совокупностью $\{f_1, f_2, \ldots, f_s\}$. Поэтому можно писать $\mathfrak{A} = C[f_1, f_2, \ldots, f_s]$.

Сопоставим теперь каждой формуле $\mathfrak{A}(x_1, \ldots, x_n)$ над $\mathfrak{P}$ функцию $f(x_1, \ldots, x_n)$ из P_2, опираясь на индуктивное определение формул.

а) Базис индукции. Если $\mathfrak{A}(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$, где $f \in \mathfrak{P}$, то формуле $\mathfrak{A}(x_1, \ldots, x_n)$ сопоставим функцию $f(x_1, \ldots, x_n)$.

б) Индуктивный переход. Пусть $\mathfrak{A}(x_1, \ldots, x_n) = f_0(A_1, \ldots, A_m)$, где A_i $(i = 1, \ldots, m)$ является либо формулой над $\mathfrak{P}$, либо символом переменной $x_{j(i)}$. Тогда по предположению индукции A_i сопоставлена либо функция f_i из P_2, либо тождественная функция $f_i = x_{j(i)}$. Сопоставим формуле $\mathfrak{A}(x_1, \ldots, x_n)$ функцию $f(x_1, \ldots, x_n) = f_0(f_1, \ldots, f_m)$.

Если функция f соответствует формуле $\mathfrak{A}$, то говорят также, что формула $\mathfrak{A}$ *реализует функцию* f. Поскольку функции рассматриваются с точностью до фиктивных переменных, мы считаем, что формула $\mathfrak{A}$ реализует и любую функцию, равную f.

Замечание 1. Если функция $f(x_1, \ldots, x_n)$, реализуемая формулой $\mathfrak{A}(x_1, \ldots, x_n)$, имеет несущественную переменную x_i, то при $n > 1$ переменную x_i можно удалить, заменив функцию f равной ей функцией f', а формулу $\mathfrak{A}$ — формулой $\mathfrak{A}'$, получающейся из $\mathfrak{A}$ в результате отождествления переменной x_i с любой из оставшихся переменных. Очевидно, что $\mathfrak{A}'$ является формулой над $\mathfrak{P}$ и реализует функцию f'.

Функцию f, соответствующую формуле $\mathfrak{A}$, будем называть *суперпозицией* функций из $\mathfrak{P}$, а процесс получения функции f из $\mathfrak{P}$ будем называть *операцией суперпозиции.*

Пример 3. Пусть $f_1(x_1, x_2)$, $f_2(x_1, x_2, x_3)$ и $f_3(x_1, x_2, x_3)$ — функции, соответствующие формулам из примера 1.

1) Формула $(((x_1x_2)+x_1)+x_2)$ строится за три шага. Мы имеем следующие подформулы:

$$(x_1x_2), \quad ((x_1x_2)+x_1), \quad (((x_1x_2)+x_1)+x_2).$$

В табл. 4 приводятся соответствующие им функции, которые вычисляются с использованием табл. 3 и правила б). Последний столбец определяет функцию $f_1(x_1, x_2)$; очевидно, что $f_1(x_1, x_2) = \max(x_1, x_2)$.

Таблица 4

x_1	x_2	(x_1x_2)	$((x_1x_2)+x_1)$	$(((x_1x_2)+x_1)+x_2)$
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	0	1

2) Функцию $f_2(x_1, x_2, x_3)$ мы будем строить несколько иным путем (также вытекающим из определения). Для каждого набора $(\sigma_1, \sigma_2, \sigma_3)$, используя табл. 2 и 3, найдем

$$f_2(\sigma_1, \sigma_2, \sigma_3) = (\overline{\sigma_1}(\sigma_2 + \sigma_3))$$

(см. табл. 5).

Таблица 5

x_1	x_2	x_3	$f_2(x_1, x_2, x_3)$	$f_3(x_1, x_2, x_3)$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

3) Для нахождения функции $f_3(x_1, x_2, x_3)$ будем, опираясь на табл. 2 и 3, искать те наборы, на которых формула

$$\overline{\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}}$$

обращается в 1. Очевидно, что это равносильно нахождению случаев, при которых формула

$$\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}$$

равна 0. Последнее имеет место при $x_1 = 0$ и в тех случаях, когда $[(x_2 \to x_3)(x_3 \to x_2)]$ обращается в 0. Наконец, формула $[(x_2 \to x_3)(x_3 \to x_2)]$ обращается в 0, когда по крайней мере одна из формул $(x_2 \to x_3)$, $(x_3 \to x_2)$ обращается в 0, что имеет место при $x_2 = 1$, $x_3 = 0$ или при $x_2 = 0$, $x_3 = 1$. Таким образом, $f_3(x_1, x_2, x_3)$ равна 1 на наборах $(0, 0, 1)$ и $(0, 1, 0)$ (см. табл. 5).

Замечание 2. Пункт б) в определении формул можно заменить на пункт, использующий две более простые операции.

1) *Операция подстановки переменных.* Пусть

$$S = \begin{pmatrix} x_1 & \dots & x_n \\ x_{i_1} & \dots & x_{i_n} \end{pmatrix}$$

— подстановка переменных (x_{i_ν} не обязана отличаться от x_{i_μ} при $\nu \neq \mu$). Эта подстановка позволяет произвести подстановку переменных у функции $f(x_1, \ldots, x_n)$ и получить в результате функцию $f(x_{i_1}, \ldots, x_{i_n})$. Очевидно, подстановка переменных включает в себя переименование переменных, перестановку переменных и отождествление переменных.

2) *Операция бесповторной подстановки функций.* Она позволяет строить выражения $f(A_1, \ldots, A_m)$, где A_i — либо формула, либо переменная из U, причем хотя бы одно из A_i отлично от переменной, а множества переменных, входящих в A_i и A_j, не пересекаются.

Очевидно, что всякая формула над $\mathfrak{P}$ может быть получена из функций, принадлежащих $\mathfrak{P}$, путем применения сначала операции бесповторной подстановки функций (многократной), а затем операции подстановки переменных (причем однократной).

Замечание 3. Если $\mathfrak{P}$ содержит тождественную функцию, то формулировка пункта б) в определении формул и функций, реализуемых формулами, упрощается: нужно предполагать, что все A_i являются формулами над $\mathfrak{P}$.

Введенный нами язык формул удобен для записи функций алгебры логики, описывающих различные условия или высказывания. Для иллюстрации рассмотрим два примера. В них используются высказывания вида «имеет место x» или просто «x», а это в свою очередь означает, что при данных условиях x истинно или равно 1.

Пример 4. *Сложение n-разрядных двоичных чисел.* Мы исходим из обычного алгоритма сложения «столбиком»

$$\begin{array}{r} x_n \ldots x_2 x_1 \\ + \\ y_n \ldots y_2 y_1 \\ \hline z_{n+1} z_n \ldots z_2 z_1 \end{array}.$$

Требуется выразить значения разрядов суммы через значения разрядов слагаемых. Для решения этого вопроса рассматривают вспомогательные величины w_n, w_{n-1}, ..., w_1, где w_i обозначает результат переноса из i-го разряда в $(i+1)$-й разряд. Эти параметры появляются в упомянутом алгоритме.

Ясно, что тогда

$$z_i = ((x_i + y_i) + w_{i-1})$$
$$(w_0 = 0, \quad x_{n+1} = y_{n+1} = 0, \quad i = 1, \ldots, n+1).$$

Величина w_i определяется условием переноса из i-го разряда в $(i+1)$-й разряд: «перенос в $(i+1)$-й разряд имеет место тогда и только тогда, когда по крайней мере две из трех величин x_i, y_i, w_{i-1} равны 1». Это высказывание более подробно можно сформулировать так: «x_i и y_i» или «x_i и w_{i-1}» или «y_i и w_{i-1}».

Если теперь заменить союзы «и» и «или» символами & и $\vee$, то получим следующую формулу для w_i:

$$w_i = \{[(x_i \,\&\, y_i) \vee (x_i \,\&\, w_{i-1})] \vee (y_i \,\&\, w_{i-1})\},$$
$$(i = 1, \ldots, n).$$

Пример 5. *Задача о вызове свободного лифта.* Пусть в подъезде имеется три лифта, обслуживающих n этажей. На каждом этаже имеется устройство, которое позволяет при нажатии кнопки вызывать ближайший свободный лифт. Спрашивается, как можно на логическом языке записать условие вызова i-го лифта ($i = 1, 2, 3$)? Мы рассмотрим эту задачу для случая вызова лифта на первом этаже.

Для описания исходной информации введем $3n$ аргументов

$$x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n,$$

где $x_i = 1$ тогда и только тогда, когда 1-й лифт находится на i-м этаже и свободен; $y_i = 1$ тогда и только тогда, когда 2-й лифт находится на i-м этаже и свободен; $z_i = 1$ тогда и только тогда, когда 3-й лифт находится на i-м этаже и свободен.

Обозначим через

$$f_1(x_1, y_1, z_1, \ldots, x_n, y_n, z_n), \ldots, f_{111}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n)$$

функции, равные 1 тогда и только тогда, когда вызывается лифт с номером соответственно 1, 2, 3. Условие вызова 1-го лифта, или функция f_1, характеризуется тем, что «1-й лифт свободен и нет свободных лифтов, расположенных на более низком этаже, чем 1-й лифт». Это высказывание можно выразить подробнее следующим образом: «1-й лифт вызывается тогда и только тогда, когда 1-й лифт находится на 1-м этаже и свободен или на 1-м этаже нет свободных лифтов с номерами 2 и 3, и в этом случае 1-й лифт находится на 2-м этаже

и свободен, или на 2-м этаже нет свободных лифтов с номерами 2 и 3 и здесь, в свою очередь...» Запишем это высказывание через высказывания $x_1, y_1, z_1, \ldots$ $\ldots, x_n, y_n, z_n$: «1-й лифт вызывается тогда и только тогда, когда x_1, или когда «не y_1 и не z_1», и в этом случае x_2 или «не y_2 и не z_2» и здесь, в свою очередь...». Теперь нетрудно получить формулу для f_{I}, если заменить союзы «и» и «или» на & и $\vee$, частицу «не» на $^-$ и расставить скобки в соответствии с соединительными словами:

$$f_{\text{I}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = \{x_1 \vee \{(\bar{y}_1 \,\&\, \bar{z}_1) \,\&\, [x_2 \vee [(\bar{y}_2 \,\&\, \bar{z}_2) \,\&\, (\ldots)]]\}\}.$$

Аналогичные формулы получаются для функций f_{II} и f_{III}:

$$f_{\text{II}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = \{y_1 \vee \{\bar{x}_1 \,\&\, \bar{z}_1) \,\&\, [y_2 \vee [(\bar{x}_2 \,\&\, \bar{z}_2) \,\&\, (\ldots)]]\}\},$$

$$f_{\text{III}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = \{z_1 \vee \{\bar{x}_1 \,\&\, \bar{y}_1) \,\&\, [z_2 \vee [(\bar{x}_2 \,\&\, \bar{y}_2) \,\&\, (\ldots)]]\}\}.$$

Таким образом, язык формул представляет известный интерес.

§ 3. Эквивалентность формул. Свойства элементарных функций. Принцип двойственности

Мы видели, что каждой формуле над $\mathfrak{P}$ соответствует функция алгебры логики, причем различным формулам могут соответствовать равные функции (см., в частности, пример 6).

Определение. Формулы $\mathfrak{A}$ и $\mathfrak{B}$ над $\mathfrak{P}$ называются *эквивалентными*, если соответствующие им функции $f_{\mathfrak{A}}$ и $f_{\mathfrak{B}}$ равны, т. е. $f_{\mathfrak{A}} = f_{\mathfrak{B}}$. Запись $\mathfrak{A} = \mathfrak{B}$ будет означать, что формулы $\mathfrak{A}$ и $\mathfrak{B}$ эквивалентны.

Пример 6.

1) $0 = (x \,\&\, \bar{x})$,

2) $(\bar{x}_1(x_2 + x_3)) = \overline{\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}}$,

3) $(x \to y) = (\bar{y} \to \bar{x})$.

Приведем список эквивалентностей (тождеств), характеризующих свойства некоторого множества элементарных функций (главным образом множества $\{0, 1, \bar{x}, (x_1 \,\&\, x_2), (x_1 \vee x_2)\}$).

Обозначим через $(x_1 \circ x_2)$ любую из функций $(x_1 \& x_2)$, $(x_1 \vee x_2)$, $(x_1 + x_2)$. Существенно только, чтобы символ $\circ$ в тождестве всюду имел один и тот же смысл.

1) Функция $(x_1 \circ x_2)$ обладает свойством ассоциативности:

$$((x_1 \circ x_2) \circ x_3) = (x_1 \circ (x_2 \circ x_3)).$$

2. Функция $(x_1 \circ x_2)$ обладает свойством коммутативности:

$$(x_1 \circ x_2) = (x_2 \circ x_1).$$

3. Для конъюнкции и дизъюнкции выполняются дистрибутивные законы:

$$((x_1 \vee x_2) \& x_3) = ((x_1 \& x_3) \vee (x_2 \& x_3)),$$
$$((x_1 \& x_2) \vee x_3) = ((x_1 \vee x_3) \& (x_2 \vee x_3)).$$

4. Имеют место следующие соотношения между отрицанием, конъюнкцией и дизъюнкцией:

$$\bar{\bar{x}} = x, \quad \overline{(x_1 \& x_2)} = (\bar{x}_1 \vee \bar{x}_2), \quad \overline{(x_1 \vee x_2)} = (\bar{x}_1 \& \bar{x}_2).$$

5. Выполняются следующие свойства конъюнкции и дизъюнкции:

$$(x \& x) = x, \quad (x \vee x) = x,$$
$$(x \& \bar{x}) = 0, \quad (x \vee \bar{x}) = 1,$$
$$(x \& 0) = 0, \quad (x \vee 0) = x,$$
$$(x \& 1) = x, \quad (x \vee 1) = 1.$$

Тождества легко могут быть проверены путем сопоставления функций, соответствующих правой и левой частям тождеств.

Замечания. 1. С целью упрощения записи формул мы условимся, что операция $\&$ *сильнее* операции $\vee$, т. е. если нет скобок, то сначала выполняется операция $\&$, а потом $\vee$. Например, запись $(x_1 \& x_2 \vee x_3)$ означает $((x_1 \& x_2) \vee x_3)$.

2. В силу закона ассоциативности для $(x_1 \circ x_2)$ можно вместо формул $((x_1 \circ x_2) \circ x_3)$, $(x_1 \circ (x_2 \circ x_3))$ пользоваться выражением $(x_1 \circ x_2 \circ x_3)$, которое не является формулой, но может быть превращено в нее путем расстановки скобок, причем функциональные свойства не меняются, как бы мы ни расставляли скобки.

3. В формулах, у которых внешняя функция является либо конъюнкцией, либо дизъюнкцией, либо сложе-

нием по mod 2, либо импликацией, либо функцией Шеффера, внешние скобки опускаются, например, пишем $x_1 \to x_2$ вместо $(x_1 \to x_2)$; опускаются также скобки у выражения, над которым стоит знак $\bar{}$, например, пишем $\overline{x_1 \to x_2}$ вместо $\overline{(x_1 \to x_2)}$.

В дальнейшем будем употреблять следующие обозначения:

$$\&_{i=1}^{s} x_i = x_1 \& x_2 \& \ldots \& x_s,$$

$$\bigvee_{i=1}^{s} x_i = x_1 \vee x_2 \vee \ldots \vee x_s.$$

Эти записи имеют смысл также и при $s = 1$.

Удобно сформулировать ряд правил, вытекающих из пунктов 2 и 5 списка тождеств элементарных функций.

Если в логическом произведении один из множителей равен 0, *то и логическое произведение равно* 0.

Если в логическом произведении, содержащем не менее двух множителей, имеется множитель, равный 1, *то этот множитель можно зачеркнуть.*

Если в логической сумме, содержащей не менее двух слагаемых, имеется слагаемое, равное 0, *то это слагаемое можно зачеркнуть.*

Если в логической сумме одно из слагаемых равно 1, *то и логическая сумма равна* 1.

В дальнейшем, используя замечания 1, 2 и 3, мы будем употреблять не формулы, а выражения, отличающиеся от формул тем, что в них кое-где опущены скобки. Эти выражения мы также иногда будем называть формулами.

Очевидно, что *если* $\mathfrak{A}'$ *— подформула формулы* $\mathfrak{A}$ *и если заменить любое из ее вхождений на эквивалентную формулу* $\mathfrak{B}'$, *то формула* $\mathfrak{A}$ *перейдет в формулу* $\mathfrak{B}$, *которая будет эквивалентна* $\mathfrak{A}$.

Этот принцип вместе с тождествами для элементарных функций, к которым присоединяются все тождества, получаемые подстановкой вместо переменных x, x_1, x_2, x_3 любых формул, позволяет осуществлять эквивалентные преобразования и, тем самым, получать новые тождества.

Пример 7.

$$x_1 \vee x_1x_2 = x_1 \cdot 1 \vee x_1x_2 = x_1(x_2 \vee \bar{x}_2) \vee x_1x_2 =$$
$$= x_1x_2 \vee x_1\bar{x}_2 \vee x_1x_2 = x_1x_2 \vee x_1x_2 \vee x_1\bar{x}_2 =$$
$$= x_1x_2 \vee x_1\bar{x}_2 = x_1(x_2 \vee \bar{x}_2) = x_1 \cdot 1 = x_1.$$

Таким образом, $x_1 \vee x_1x_2 = x_1$, т. е. получаем правило *поглощения произведения x_1x_2 множителем x_1*.

Существует еще один способ для получения тождеств. Он основан на использовании так называемого принципа двойственности.

Определение. Функция $[f(x_1, \ldots, x_n)]^*$, равная $\bar{f}(\bar{x}_1, \ldots, \bar{x}_n)$, называется *двойственной функцией к функции* $f(x_1, \ldots, x_n)$.

Очевидно, что таблица для двойственной функции (при выбранном порядке наборов) получается из таблицы для функции $f(x_1, \ldots, x_n)$ инвертированием (т. е. заменой 0 на 1 и 1 на 0) столбца функции и его переворачиванием (см. табл. 6).

Таблица 6

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	$[f(x_1,x_2,x_3)]^*$
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	0

Поскольку $[f(x_{i_1}, \ldots, x_{i_n})]^* = \bar{f}(\bar{x}_{i_1}, \ldots, \bar{x}_{i_n})$, то функция $[f(x_{i_1}, \ldots, x_{i_n})]^*$ определяет то же отображение, что и $[f(x_1, \ldots, x_n)]^*$. Обозначим это отображение через f^*. Тогда

$$[f(x_1, \ldots, x_n)]^* = f^*(x_1, \ldots, x_n).$$

Легко видеть, что среди функций 0, 1, x, $\bar{x}$, $x_1 \& x_2$, $x_1 \vee x_2$

функция 0 двойственна 1,
функция 1 двойственна 0,
функция x двойственна x,
функция $\bar{x}$ двойственна $\bar{x}$,
функция $x_1 \& x_2$ двойственна $x_1 \vee x_2$,
функция $x_1 \vee x_2$ двойственна $x_1 \& x_2$.

Из определения двойственности вытекает, что

$$f^{**} = (f^*)^* = f,$$

т. е. *функция f является двойственной к f^** (свойство взаимности).

Пусть $f(x_1, \ldots, x_n)$ выражена формулой $\mathfrak{A}$. Спрашивается, какой вид имеет формула $\mathfrak{B}$, реализующая $f^*(x_1, \ldots, x_n)$?

Обозначим через $x_1, \ldots, x_n$ все различные символы переменных, встречающиеся в множествах

$$(x_{11}, \ldots, x_{1p_1}), \ldots, (x_{m1}, \ldots, x_{mp_m}).$$

Теорема 2. *Если*

$$\Phi(x_1, \ldots, x_n) = f(f_1(x_{11}, \ldots, x_{1p_1}), \ldots, f_m(x_{m1}, \ldots, x_{mp_m})),$$

то

$$\Phi^*(x_1, \ldots, x_n) = f^*(f_1^*(x_{11}, \ldots, x_{1p_1}), \ldots, f_m^*(x_{m1}, \ldots, x_{mp_m})).$$

Доказательство.

$$\Phi^*(x_1, \ldots, x_n) = \overline{\Phi}(\overline{x}_1, \ldots, \overline{x}_n) =$$
$$= \bar{f}(f_1(\overline{x}_{11}, \ldots, \overline{x}_{1p_1}), \ldots, f_m(\overline{x}_{m1}, \ldots, \overline{x}_{mp_m})) =$$
$$= \bar{f}(\bar{f}_1(\overline{x}_{11}, \ldots, \overline{x}_{1p_1}), \ldots, \bar{f}_m(\overline{x}_{m1}, \ldots, \overline{x}_{mp_m})) =$$
$$= \bar{f}(\bar{f}_1^*(x_{11}, \ldots, x_{1p_1}), \ldots, \bar{f}_m^*(x_{m1}, \ldots, x_{mp_m})) =$$
$$= f^*(f_1^*(x_{11}, \ldots, x_{1p_1}), \ldots, f_m^*(x_{m1}, \ldots, x_{mp_m})).$$

Из теоремы вытекает

Принцип двойственности. *Если формула $\mathfrak{A} = C[f_1, \ldots, f_s]$ реализует функцию $f(x_1, \ldots, x_n)$, то формула $C[f_1^*, \ldots, f_s^*]$, т. е. формула, полученная из $\mathfrak{A}$ заменой функций $f_1, \ldots, f_s$ соответственно на $f_1^*, \ldots, f_s^*$, реализует функцию $f^*(x_1, \ldots, x_n)$.*

Эту формулу мы будем называть *формулой, двойственной к $\mathfrak{A}$*, и обозначать через $\mathfrak{A}^*$. Таким образом,

$$\mathfrak{A}^* = C[f_1^*, \ldots, f_s^*].$$

Для формул над множеством $\mathfrak{P} = \{0, 1, \bar{x}, x_1 \& x_2, x_1 \vee x_2\}$ принцип двойственности может быть сформулирован так: *для получения формулы $\mathfrak{A}^*$, двойственной к формуле $\mathfrak{A}$, нужно в формуле $\mathfrak{A}$ всюду заменить*

$$0 \text{ на } 1, \quad 1 \text{ на } 0, \quad \& \text{ на } \vee, \quad \vee \text{ на } \&.$$

Или, если $\mathfrak{A} = C[0, 1, \bar{x}, x_1 \& x_2, x_1 \vee x_2]$, *то*

$$\mathfrak{A}^* = C[1, 0, \bar{x}, x_1 \vee x_2, x_1 \& x_2].$$

Пример 8.

1) $\mathfrak{A}_1(x_1, x_2) = x_1 \& x_2$, $\mathfrak{A}_1^*(x_1, x_2) = x_1 \vee x_2$;

2) $\mathfrak{A}_2(x_1, x_2) = x_1 x_2 \vee \bar{x}_1 \bar{x}_2$, $\mathfrak{A}_2^*(x_1, x_2) = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2)$.

Из принципа двойственности вытекает, что если

$$\mathfrak{A}(x_1, \ldots, x_n) = \mathfrak{B}(x_1, \ldots, x_n),$$

то

$$\mathfrak{A}^*(x_1, \ldots, x_n) = \mathfrak{B}^*(x_1, \ldots, x_n).$$

Пример 9. Из тождества $\overline{x_1 \& x_2} = \bar{x}_1 \vee \bar{x}_2$ вытекает тождество $x_1 \vee x_2 = \bar{x}_1 \& \bar{x}_2$.

Принцип двойственности позволяет почти в два раза сокращать усилия на вывод тождеств при рассмотрении свойств элементарных функций. Другие применения принципа двойственности будут даны ниже.

§ 4. Разложение булевых функций по переменным. Совершенная дизъюнктивная нормальная форма

Говоря о языке формул, мы сознательно не касались весьма важного вопроса. Если в качестве $\mathfrak{P}$ допустить некоторый запас элементарных функций, то всякая ли функция алгебры логики может быть выражена в виде формулы? Ближайшие рассмотрения направлены на решение этого вопроса.

Введем обозначение

$$x^\sigma = x\sigma \vee \bar{x}\bar{\sigma},$$

где σ — параметр, равный либо 0, либо 1. Очевидно, что

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

Легко видеть, что $x^\sigma = 1$ тогда и только тогда, когда $x = \sigma$, т. е. значение «основания» равно значению «показателя».

Теорема 3 (о разложении функций по переменным). *Каждую функцию алгебры логики* $f(x_1, \ldots, x_n)$ *при любом* m $(1 \leqslant m \leqslant n)$ *можно представить в*

следующей форме:

$$f(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) =$$

$$= \bigvee_{(\sigma_1,\ldots,\sigma_m)} x_1^{\sigma_1} \& \ldots \& x_m^{\sigma_m} \& f(\sigma_1, \ldots, \sigma_m, x_{m+1}, \ldots, x_n), \quad (*)$$

где дизъюнкция берется по всевозможным наборам значений переменных $x_1, \ldots, x_m$.

Это представление называется *разложением функции по* m *переменным* $x_1, \ldots, x_m$.

Доказательство. Рассмотрим произвольный набор значений переменных $(\alpha_1, \ldots, \alpha_n)$ и покажем, что левая и правая части соотношения $(*)$ принимают на нем одно и то же значение. Левая часть дает $f(\alpha_1, \ldots, \alpha_n)$. Правая —

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

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

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

В качестве следствий получаем два специальных случая разложения.

1) Разложение по переменной:

$$f(x_1, \ldots, x_{n-1}, x_n) =$$

$$= x_n \& f(x_1, \ldots, x_{n-1}, 1) \vee \bar{x}_n \& f(x_1, \ldots, x_{n-1}, 0).$$

Функции $f(x_1, \ldots, x_{n-1}, 0)$ и $f(x_1, \ldots, x_{n-1}, 1)$ называются *компонентами разложения*. Данное разложение полезно, когда какие-либо свойства булевых функций устанавливаются по индукции.

2) Разложение по всем n переменным:

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

При $f(x_1, \ldots, x_n) \not\equiv 0$ оно может быть преобразовано:

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

$$= \bigvee_{\substack{(\sigma_1,\ldots,\sigma_n) \\ f(\sigma_1,\ldots,\sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

В результате окончательно получим

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

Такое разложение носит название *совершенной дизъюнктивной нормальной формы* (совершенной д. н. ф.).

Непосредственно к понятию совершенной д. н. ф. примыкает следующая теорема.

Теорема 4. *Каждая функция алгебры логики может быть выражена в виде формулы через отрицание, конъюнкцию и дизъюнкцию.*

Доказательство. 1) Пусть $f(x_1, \ldots, x_n) \equiv 0$. Тогда, очевидно,

$$f(x_1, \ldots, x_n) = x_1 \& \bar{x}_1.$$

2) Пусть $f(x_1, \ldots, x_n) \not\equiv 0$. Представим ее в совершенной д. н. ф.:

$$f(x_1, \ldots, x_n) = \bigvee_{\substack{(\sigma_1,\ldots,\sigma_n) \\ f(\sigma_1,\ldots,\sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Таким образом, в обоих случаях функция f выражается в виде формулы через отрицание, конъюнкцию и дизъюнкцию.

Итак, оказалось, что любую булеву функцию можно задать формулой над $\mathfrak{P}$, взяв в качестве $\mathfrak{P}$ множество, состоящее из трех функций: отрицания, конъюнкции и дизъюнкции.

Данная теорема носит конструктивный характер, так как она позволяет для каждой функции фактически построить формулу, ее реализующую (в виде совершенной д. н. ф.): в таблице для функции $f(x_1, \ldots, x_n)$ $(f \not\equiv 0)$ отмечаем все строки $(\sigma_1, \ldots, \sigma_n)$, в которых $f(\sigma_1, \ldots, \sigma_n) = 1$; для каждой такой строки образуем логическое произведение

$$x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n},$$

и затем все полученные конъюнкции соединяем знаком дизъюнкции.

Пример 10. 1) Найти совершенную д. н. ф. для функции $x_1 \to x_2$.

Мы имеем три набора (0, 0), (0, 1) и (1, 1), на которых эта функция равна 1 (см. табл. 3). Поэтому

$$x_1 \to x_2 = x_1^0 \,\&\, x_2^0 \vee x_1^0 \,\&\, x_2^1 \vee x_1^1 \,\&\, x_2^1 =$$

$$= \bar{x}_1 \,\&\, \bar{x}_2 \vee \bar{x}_1 \,\&\, x_2 \vee x_1 \,\&\, x_2.$$

2) Найти совершенную д. н. ф. для функции, заданной табл. 7.

Имеем

$$f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3.$$

Совершенная д. н. ф. есть выражение типа ΣΠ, т. е. логическая сумма произведений $x_i^{\sigma_i}$. Спрашивается, нельзя ли для булевых функций получить разложение типа

Таблица 7

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

ΠΣ? Покажем, что при $f \not\equiv 1$ это возможно, для чего разложим функцию f^* (очевидно, $f^* \not\equiv 0$) в совершенную д. н. ф.:

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

Возьмем тождество для двойственных формул

$$f^{**}(x_1, \ldots, x_n) = \bigwedge_{\substack{(\sigma_1,\ldots,\sigma_n) \\ f^*(\sigma_1,\ldots,\sigma_n)=1}} (x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}).$$

Левая часть есть $f(x_1, \ldots, x_n)$, а правая может быть преобразована далее:

$$\&_{\substack{(\sigma_1,\ldots,\sigma_n)\\ f^*(\sigma_1,\ldots,\sigma_n)=1}} (x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}) = \&_{\substack{(\sigma_1,\ldots,\sigma_n)\\ f(\bar{\sigma}_1,\ldots,\bar{\sigma}_n)=0}} (x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}) =$$

$$= \&_{\substack{(\sigma_1,\ldots,\sigma_n)\\ f(\sigma_1,\ldots,\sigma_n)=0}} (x_1^{\bar{\sigma}_1} \vee \ldots \vee x_n^{\sigma_n}).$$

Таким образом, получаем разложение

$$f(x_1, \ldots, x_n) = \&_{\substack{(\sigma_1,\ldots,\sigma_n)\\ f(\sigma_1,\ldots,\sigma_n)=0}} (x_1^{\bar{\sigma}_1} \vee \ldots \vee x_n^{\bar{\sigma}_n}).$$

Это выражение носит название *совершенной конъюнктивной нормальной формы* (совершенной к. н. ф.).

Пример 11. 1). Построить совершенную к. н. ф. для функции $x_1 \to x_2$.

Имеем

$$x_1 \to x_2 = x_1^{\bar{1}} \vee x_2^{\bar{0}} = \bar{x}_1 \vee x_2.$$

2) Построить совершенную к. н. ф. для функции $f(x_1, x_2, x_3)$, заданной табл. 7.

Имеем

$$f(x_1, x_2, x_3) = (x_1 \vee \bar{x}_2 \vee x_3)(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \times$$
$$\times (\bar{x}_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3).$$

Итак, в качестве средства для задания булевых функций наряду с таблицами можно использовать язык формул над множеством функций, состоящим из отрицания, конъюнкции и дизъюнкции. Поскольку табличный язык по громоздкости примерно эквивалентен языку совершенных д. н. ф. (совершенных к. н. ф), то можно утверждать, что язык формул, использующий отрицания, конъюнкции и дизъюнкции, не хуже языка таблиц. Можно показать, что на самом деле он существенно лучше табличного языка. Для пояснения рассмотрим функцию

$$f(x_1, \ldots, x_{20}) = x_1 \vee \ldots \vee x_{20}.$$

Данная формула в правой части насчитывает 39 символов (20 символов переменных и 19 символов $\vee$), таблица для $f(x_1, \ldots, x_{20})$ содержит 2^{20}, т. е. более миллиона строк.

§ 5. Полнота и замкнутость

Выше мы видели, что всякая функция алгебры логики может быть выражена в виде формулы через элементарные функции $\bar{x}$, $x_1 \& x_2$ и $x_1 \vee x_2$. В связи с этим возникает вопрос, в какой мере является случайным наличие таких систем элементарных функций? Сейчас мы не собираемся давать исчерпывающего ответа на поставленный вопрос, а лишь покажем, что такого рода свойством обладают и некоторые другие системы элементарных функций.

Определение. Система функций $\{f_1, f_2, \ldots, f_n, \ldots\}$ из P_2 называется (функционально) *полной*, если любая булева функция может быть записана в виде формулы через функции этой системы.

Рассмотрим примеры полных систем.

1. Система P_2 — множество всех булевых функций — является полной системой.

2. Система $\mathfrak{P} = \{\bar{x}, x_1 \& x_2, x_1 \vee x_2\}$ представляет полную систему.

Ясно, что не каждая система является полной, например, система $\mathfrak{P} = \{0, 1\}$ не полная. Следующая теорема позволяет сводить вопрос о полноте одних систем к вопросу о полноте других систем.

Теорема 5. *Пусть даны две системы функций из* P_2:

$$\mathfrak{P} = \{f_1, f_2, \ldots\}, \tag{I}$$

$$\mathfrak{D} = \{g_1, g_2, \ldots\}, \tag{II}$$

относительно которых известно, что система I *полна и каждая ее функция выражается в виде формулы через функции системы* II. *Тогда система* II *является полной.*

Доказательство. Пусть h — произвольная функция из P_2. В силу полноты системы I можно выразить h формулой над $\mathfrak{P}$, т. е.

$$h = C[f_1, f_2, \ldots, f_n, \ldots]$$

(в скобках мы выписываем все функции системы $\mathfrak{P}$, хотя в формуле фактически встречается лишь конечное их число). По условию теоремы

$$f_1 = C_1[g_1, g_2, \ldots],$$

$$f_2 = C_2[g_1, g_2, \ldots],$$

$$\ldots\ldots\ldots$$

Поэтому мы можем в формуле $C[f_1, f_2, \ldots]$ исключить вхождения функций $f_1, f_2, \ldots$, заменив их формулами над $\mathfrak{Q}$*). Это можно записать так:

$$C[f_1, f_2, \ldots] = C[C_1[g_1, g_2, \ldots], C_2[g_1, g_2, \ldots], \ldots].$$

Последнее выражение определяет формулу над $\mathfrak{Q}$ со строением C'. Мы получаем

$$C[C_1[g_1, g_2, \ldots], C_2[g_1, g_2, \ldots], \ldots] = C'[g_1, g_2, \ldots],$$

или, окончательно,

$$h = C'[g_1, g_2, \ldots],$$

т. е. мы выразили h в виде формулы над $\mathfrak{Q}$. Теорема доказана.

Опираясь на эту теорему, можно установить полноту еще ряда систем и тем самым расширить список примеров полных систем.

3. Система $\mathfrak{P} = \{\bar{x}, x_1 \& x_2\}$ является полной. Для доказательства возьмем за систему I систему 2 (стр. 30), а за систему II — систему 3 и используем тождество

$$x_1 \vee x_2 = \overline{\bar{x}_1 \& \bar{x}_2},$$

которое вытекает из свойств 4 элементарных функций.

4. Система $\mathfrak{P} = \{\bar{x}, x_1 \vee x_2\}$ является полной.

Этот факт доказывается либо аналогично предыдущему, либо через принцип двойственности.

5. Система $\mathfrak{P} = \{x_1/x_2\}$ является полной.

Для доказательства за систему I возьмем систему 3, а за систему II — систему 5. Легко видеть, что

$$x_1/x_1 = \bar{x}_1, \quad (x_1/x_2)/(x_1/x_2) = \overline{x_1/x_2} = x_1 \& x_2.$$

6. Система $\mathfrak{P} = \{0, 1, x_1x_2, x_1 + x_2\}$ является полной.

---

*) Не всегда эти замены можно произвести все одновременно, так как при замене, вообще говоря, могут возникнуть новые вхождения символов $f_1, f_2, \ldots$ Например, если

$$f_1 = x_1 \& x_2, \quad g_1 = x_1 \vee x_2, \quad g_2 = x_1 + x_2, \quad h = x_1 \& (x_2 \& x_3),$$

то

$$f_1 = x_1 \& x_2 = x_1 + x_2 + (x_1 \vee x_2),$$

$$h = x_1 \& (x_2 \& x_3) = x_1 + (x_2 \& x_3) + (x_1 \vee (x_2 \& x_3)) =$$

$$= x_1 + x_2 + x_3 + (x_2 \vee x_3) + (x_1 \vee (x_2 + x_3 + (x_2 \vee x_3))).$$

Для доказательства опять за систему I возьмем систему 3, а за систему II — систему 6. Мы имеем

$$x_1 + 1 = \bar{x}_1,$$

$$x_1 x_2 = x_1 \& x_2.$$

Так как формула, построенная из констант 0, 1 и функций $x_1 x_2$ и $x_1 + x_2$, после раскрытия скобок и несложных алгебраических преобразований переходит в полином по mod 2, то мы получаем следующую теорему, принадлежащую И. И. Жегалкину.

Теорема 6. *Каждая функция из P_2 может быть выражена при помощи полинома по* mod 2 (*полинома Жегалкина*).

Подсчитаем число полиномов Жегалкина от переменных $x_1, \ldots, x_n$, т. е. число выражений вида

$$\sum_{(i_1, \ldots, i_s)} a_{i_1 \ldots i_s} x_{i_1} \ldots x_{i_s}.$$

Число членов $x_{i_1} \ldots x_{i_s}$ равно количеству подмножеств $(i_1, \ldots, i_s)$ из n чисел $(1, \ldots, n)$, т. е. 2^n. Поскольку $a_{i_1 \ldots i_s}$ равно 0 или 1, искомое число полиномов равно 2^{2^n}, т. е. числу всех булевых функций от тех же переменных. Отсюда, как следствие, получаем *единственность представления функций посредством полиномов Жегалкина.*

Пример 12. Выразить $x_1 \vee x_2$ в виде полинома Жегалкина.

Ищем выражение для $x_1 \vee x_2$ в виде полинома с неопределенными коэффициентами:

$$x_1 \vee x_2 = a x_1 x_2 + b x_1 + c x_2 + d.$$

При $x_1 = x_2 = 0$ имеем $0 = d$,
при $x_1 = 0$, $x_2 = 1$ имеем $1 = c$,
при $x_1 = 1$, $x_2 = 0$ имеем $1 = b$,
при $x_1 = x_2 = 1$ имеем $1 = a + b + c$, т. е. $a = 1$.
Мы получаем $x_1 \vee x_2 = x_1 x_2 + x_1 + x_2$.

Из приведенных примеров видно, что существует целый ряд полных систем. Каждая из них может быть принята за множество элементарных функций. Таким образом, для задания булевых функций можно использовать различные языки формул. Какой именно из языков является более удобным, зависит от характера рассматриваемой задачи.

С понятием полноты тесно связано понятие замыкания и замкнутого класса.

Определение. Пусть $\mathfrak{M}$ — некоторое подмножество функций из P_2. *Замыканием* $\mathfrak{M}$ называется множество всех булевых функций, представимых в виде формул через функции множества $\mathfrak{M}$. Замыкание множества $\mathfrak{M}$ обозначается через $[\mathfrak{M}]$. Легко видеть, что замыкание инвариантно относительно операций введения и удаления фиктивных переменных.

Пример 13.

1) $\mathfrak{M} = P_2$. Очевидно, что $[\mathfrak{M}] = P_2$.

2) $\mathfrak{M} = \{1, x_1 + x_2\}$. Замыканием этого множества будет класс L всех линейных функций, т. е. функций, имеющих вид

$$f(x_1, \ldots, x_n) = c_0 + c_1x_1 + \ldots + c_nx_n,$$

где $c_i = 0, 1$ $(i = 0, \ldots, n)$; существенные переменные входят с коэффициентом 1, фиктивные — с коэффициентом 0.

Отметим некоторые свойства замыкания:

1) $[\mathfrak{M}] \supseteq \mathfrak{M}$;

2) $[[\mathfrak{M}]] = [\mathfrak{M}]$;

3) если $\mathfrak{M}_1 \subseteq \mathfrak{M}_2$, то $[\mathfrak{M}_1] \subseteq [\mathfrak{M}_2]$;

4) $[\mathfrak{M}_1 \cup \mathfrak{M}_2] \supseteq [\mathfrak{M}_1] \cup [\mathfrak{M}_2]$.

Определение. Класс (множество) $\mathfrak{M}$ называется (функционально) *замкнутым*, если $[\mathfrak{M}] = \mathfrak{M}$.

Пример 14.

1) Класс $\mathfrak{M} = P_2$ является замкнутым классом.

2) Класс $\mathfrak{M} = \{1, x_1 + x_2\}$ не замкнут.

3) Класс L замкнут, так как линейное выражение, составленное из линейных выражений, является линейным.

Легко видеть, что всякий класс $[\mathfrak{M}]$ будет замкнутым. Это дает возможность получать многочисленные примеры замкнутых классов.

В терминах замыкания и замкнутого класса можно дать другое определение полноты (эквивалентное исходному): $\mathfrak{M}$ — полная система, если $[\mathfrak{M}] = P_2$.

§ 6. Важнейшие замкнутые классы. Теорема о полноте

Этот параграф мы начинаем с рассмотрения некоторых важнейших замкнутых классов в P_2.

1. Обозначим через T_0 класс всех булевых функций $f(x_1, \ldots, x_n)$, сохраняющих константу 0, т. е. функций,

для которых выполнено равенство

$$f(0, \ldots, 0) = 0.$$

Заметим, что если $f \in T_0$, а f' — функция, равная f, то и $f' \in T_0$ (это достаточно проверить для функций f и f', отличающихся одной переменной).

Легко видеть, что функции 0, x, $x_1 \& x_2$, $x_1 \vee x_2$, $x_1 + x_2$ принадлежат классу T_0, а функции 1, $\bar{x}$ не входят в T_0.

Поскольку таблица для функции f из класса T_0 в первой строке содержит значение 0, то в T_0 содержится ровно $(1/2)\,2^{2^n}$ булевых функций, зависящих от переменных $x_1, \ldots, x_n$.

Покажем, что T_0 — замкнутый класс. Так как T_0 содержит тождественную функцию, то для обоснования замкнутости T_0 достаточно показать, что функция

$$\Phi = f(f_1, \ldots, f_m)$$

принадлежит T_0, если $f, f_1, \ldots, f_m$ принадлежат классу T_0. Последнее вытекает из цепочки равенств

$$\Phi(0, \ldots, 0) = f(f_1(0, \ldots, 0), \ldots, f_m(0, \ldots, 0)) = \\ = f(0, \ldots, 0) = 0.$$

2. Обозначим через T_1 класс всех булевых функций $f(x_1, \ldots, x_n)$, сохраняющих константу 1, т. е. функций, для которых выполнено равенство

$$f(1, \ldots, 1) = 1.$$

Очевидно, что класс T_1 вместе с любой функцией содержит и любую равную ей функцию.

Легко видеть, что функции 1, x, $x_1 \& x_2$, $x_1 \vee x_2$ принадлежат классу T_1, а функции 0, $\bar{x}$ не входят в T_1.

В силу того, что класс T_1 состоит из функций, двойственных функциям из класса T_0 (или, как мы будем говорить, что класс T_1 двойствен классу T_0), нетрудно перенести результаты о классе T_0 на класс T_1. Класс T_1 содержит $(1/2)\,2^{2^n}$ функций, зависящих от переменных $x_1, \ldots, x_n$, и является замкнутым классом.

3. Обозначим через S класс всех самодвойственных функций, т. е. функций f из P_2 таких, что $f^* = f$.

Как и выше, нетрудно проверить, что добавление равных функций не выводит за пределы класса S.

Очевидно, что самодвойственными функциями будут x, $\bar{x}$. Менее тривиальным примером самодвойственной

функции является функция $h(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee \vee x_2x_3$:

$$h^*(x_1, x_2, x_3) = (x_1 \vee x_2)(x_1 \vee x_3)(x_2 \vee x_3) = \\ = x_1x_2 \vee x_1x_3 \vee x_2x_3 = h(x_1, x_2, x_3).$$

Для самодвойственной функции имеет место тождество

$$f(\bar{x}_1, \ldots, \bar{x}_n) = f(x_1, \ldots, x_n);$$

иначе говоря, на наборах $(\alpha_1, \ldots, \alpha_n)$ и $(\bar{\alpha}_1, \ldots, \bar{\alpha}_n)$, которые мы будем называть *противоположными*, самодвойственная функция принимает противоположные значения. Отсюда следует, что самодвойственная функция полностью определяется своими значениями на первой половине строк (см. табл. 1). Поэтому число самодвойственных функций, зависящих от переменных $x_1, \ldots, x_n$, равно $2^{2^{n-1}} = \sqrt{2^{2^n}}$.

Докажем теперь, что класс S замкнут. Поскольку класс S содержит тождественную функцию, достаточно показать, что функция

$$\Phi = f(f_1, \ldots, f_m)$$

является самодвойственной, если $f, f_1, \ldots, f_m$ самодвойственны. Последнее устанавливается непосредственно:

$$\Phi^* = f^*(f_1^*, \ldots, f_m^*) = f(f_1, \ldots, f_m) = \Phi.$$

Докажем теперь лемму о несамодвойственной функции.

Лемма 1. *Если $f(x_1, \ldots, x_n) \notin S$, то из нее путем подстановки функций x и $\bar{x}$ можно получить несамодвойственную функцию одного переменного, т. е. константу.*

Доказательство. Так как $f \notin S$, то найдется набор $(\alpha_1, \ldots, \alpha_n)$ такой, что

$$f(\bar{\alpha}_1, \ldots, \bar{\alpha}_n) = f(\alpha_1, \ldots, \alpha_n).$$

Рассмотрим функции $\varphi_i(x) = x^{\alpha_i}$ $(i = 1, \ldots, n)$ и положим

$$\varphi(x) = f(\varphi_1(x), \ldots, \varphi_n(x)).$$

Мы имеем

$$\varphi(0) = f(\varphi_1(0), \ldots, \varphi_n(0)) = f(0^{\alpha_1}, \ldots, 0^{\alpha_n}) =$$
$$= f(\bar{\alpha}_1, \ldots, \bar{\alpha}_n) = f(\alpha_1, \ldots, \alpha_n) = f(1^{\alpha_1}, \ldots, 1^{\alpha_n}) =$$
$$= f(\varphi_1(1), \ldots, \varphi_n(1)) = \varphi(1).$$

Лемма доказана.

4. Здесь мы будем употреблять векторную запись наборов:

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n), \quad \tilde{\beta} = (\beta_1, \ldots, \beta_n)$$

и т. п. и вместо $f(\alpha_1, \ldots, \alpha_n)$ употреблять запись $f(\tilde{\alpha})$.

Определение. Для двух наборов $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ и $\tilde{\beta} = (\beta_1, \ldots, \beta_n)$ выполнено *отношение предшествования* $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, если

$$\alpha_1 \leqslant \beta_1, \ldots, \alpha_n \leqslant \beta_n.$$

Например, $(0, 1, 0, 1) \preccurlyeq (1, 1, 0, 1)$.

Очевидно, что если $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ и $\tilde{\beta} \preccurlyeq \tilde{\gamma}$, то $\tilde{\alpha} \preccurlyeq \tilde{\gamma}$. Следует отметить, что не любые пары наборов находятся в отношении предшествования, например, наборы (0, 1) и (1, 0) в таком отношении не находятся. Таким образом, множество всех наборов длины n по отношению к операции предшествования $\preccurlyeq$ является частично упорядоченным.

Определение. Функция $f(x_1, \ldots, x_n)$ называется *монотонной*, если для любых двух наборов $\tilde{\alpha}$ и $\tilde{\beta}$ таких, что $\tilde{\alpha} \preccurlyeq \tilde{\beta}$, имеет место неравенство

$$f(\tilde{\alpha}) \leqslant f(\tilde{\beta}).$$

Нетрудно заметить, что функция, равная монотонной функции, также является монотонной.

Монотонными функциями, очевидно, будут 0, 1, x, $x_1 \& x_2$ и $x_1 \vee x_2$.

Обозначим через M множество всех монотонных функций. Покажем, что класс монотонных функций замкнут. Поскольку тождественная функция принадлежит классу M, то для установления замкнутости M достаточно показать, что функция

$$\Phi = f(f_1, \ldots, f_m)$$

является монотонной, если f, f_1, ..., f_m монотонны. Пусть $\tilde{x} = (x_1, \ldots, x_n)$, $\tilde{x}^1 = (x_{11}, \ldots, x_{1p_1})$, ..., $\tilde{x}^m = (x_{m1}, \ldots, x_{mp_m})$ — наборы переменных функций Φ, f_1, ..., f_m, причем множество переменных функции Φ состоит из тех и только тех переменных, которые встречаются у функций f_1, ..., f_m. Пусть $\tilde{\alpha}$ и $\tilde{\beta}$ — два набора длины n значений переменных $\tilde{x}$, причем $\tilde{\alpha} \preccurlyeq \tilde{\beta}$. Эти наборы определяют наборы $\tilde{\alpha}^1$, $\tilde{\beta}^1$, ..., $\tilde{\alpha}^m$, $\tilde{\beta}^m$ значений переменных $\tilde{x}^1$, ..., $\tilde{x}^m$ такие, что $\tilde{\alpha}^1 \preccurlyeq \tilde{\beta}^1$, ..., $\tilde{\alpha}^m \preccurlyeq \tilde{\beta}^m$. В силу монотонности функций f_1, ..., f_m

$$f_1(\tilde{\alpha}^1) \leqslant f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\alpha}^m) \leqslant f_m(\tilde{\beta}^m).$$

Поэтому

$$(f_1(\tilde{\alpha}^1), \ldots, f_m(\tilde{\alpha}^m)) \preccurlyeq (f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\beta}^m)),$$

и в силу монотонности f имеем

$$f(f_1(\tilde{\alpha}^1), \ldots, f_m(\tilde{\alpha}^m)) \leqslant f(f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\beta}^m)),$$

откуда получаем

$$\Phi(\tilde{\alpha}) = f(f_1(\tilde{\alpha}^1), \ldots, f_m(\tilde{\alpha}^m)) \leqslant f(f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\beta}^m)) = \Phi(\tilde{\beta}).$$

Будем называть наборы $\tilde{\alpha}$ и $\tilde{\beta}$ *соседними* (по i-й координате), если

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_n),$$
$$\tilde{\beta} = (\alpha_1, \ldots, \alpha_{i-1}, \bar{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_n).$$

Докажем теперь лемму о немонотонной функции.

Лемма 2. *Если* $f(x_1, \ldots, x_n) \notin M$, *то из нее путем подстановки констант* 0 *и* 1 *и функции* x *можно получить функцию* $\bar{x}$.

Доказательство. Сначала покажем, что найдется пара соседних наборов $\tilde{\alpha}$ и $\tilde{\beta}$ таких, что $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ и

$$f(\tilde{\alpha}) > f(\tilde{\beta}).$$

В самом деле, так как $f \notin M$, то существуют наборы $\tilde{\alpha}^1$ и $\tilde{\beta}^1$ такие, что $\tilde{\alpha}^1 \preccurlyeq \tilde{\beta}^1$ и $f(\tilde{\alpha}^1) > f(\tilde{\beta}^1)$. Если наборы

$\tilde{\alpha}^1$ и $\tilde{\beta}^1$ — соседние, то наша цель достигнута. Если $\tilde{\alpha}^1$ и $\tilde{\beta}^1$ не являются соседними, то набор $\tilde{\beta}^1$ отличается от набора $\tilde{\alpha}^1$ в t координатах, где $t>1$, причем эти t координат в наборе $\tilde{\alpha}^1$ имеют значение 0, а в наборе $\tilde{\beta}^1$ — значение 1. В силу этого между $\tilde{\alpha}^1$ и $\tilde{\beta}^1$ можно вставить $t-1$ промежуточных наборов $\tilde{\alpha}^2$, $\tilde{\alpha}^3$, ..., $\tilde{\alpha}^t$ таких, что

$$\tilde{\alpha}^1 \preccurlyeq \tilde{\alpha}^2 \preccurlyeq \ldots \preccurlyeq \tilde{\alpha}^t \preccurlyeq \tilde{\beta}^1.$$

Очевидно, что наборы, стоящие в этой цепочке рядом, будут соседними. Так как $f(\tilde{\alpha}^1)>f(\tilde{\beta}^1)$, то по крайней мере на одной из этих пар соседних наборов — обозначим их через $\tilde{\alpha}$ и $\tilde{\beta}$ $(\tilde{\alpha}\preccurlyeq\tilde{\beta})$ — будет $f(\tilde{\alpha})>f(\tilde{\beta})$. Предположим, что данные наборы имеют соседство по i-й координате и, следовательно,

$$\tilde{\alpha}=(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n),$$
$$\tilde{\beta}=(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n).$$

Рассмотрим функцию

$$\varphi(x)=f(\alpha_1, \ldots, \alpha_{i-1}, x, \alpha_{i+1}, \ldots, \alpha_n).$$

Мы имеем

$$\varphi(0)=f(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n)=f(\tilde{\alpha})>f(\tilde{\beta})=$$
$$=f(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n)=\varphi(1).$$

Последнее означает, что $\varphi(0)=1$, а $\varphi(1)=0$, т. е. $\varphi(x)=\bar{x}$. Лемма доказана.

5. Последним классом является класс L всех линейных функций.

Он, очевидно, содержит константы 0, 1, тождественную функцию x, функции $\bar{x}$, x_1+x_2 и не содержит функций $x_1 \& x_2$ и $x_1 \vee x_2$. Выше было показано, что этот класс также замкнут.

Докажем лемму о нелинейной функции.

Лемма 3. *Если $f(x_1, \ldots, x_n)\notin L$, то из нее путем подстановки констант* 0 *и* 1 *и функций вида x и $\bar{x}$, а также, быть может, путем навешивания отрицания над f, можно получить функцию $x_1 \& x_2$.*

Доказательство. Возьмем полином Жегалкина для f:

$$f(x_1, \ldots, x_n)=\sum_{(i_1, \ldots, i_s)} a_{i_1 \ldots i_s} x_{i_1} \ldots x_{i_s}.$$

В силу нелинейности полинома в нем найдется член, содержащий не менее двух множителей. Без ограничения общности можно считать, что среди этих множителей присутствуют x_1 и x_2. Тогда можно преобразовать полином следующим образом:

$$\sum_{(i_1,\ldots,i_s)} a_{i_1\ldots i_s} x_{i_1}\ldots x_{i_s} = x_1x_2f_1(x_3,\ldots,x_n) +$$

$$+ x_1f_2(x_3,\ldots,x_n) + x_2f_3(x_3,\ldots,x_n) + f_4(x_3,\ldots,x_n),$$

где в силу единственности полинома $f_1(x_3,\ldots,x_n)\not\equiv 0$.

Пусть $\alpha_3,\ldots,\alpha_n$ таковы, что $f_1(\alpha_3,\ldots,\alpha_n)=1$. Тогда

$$\varphi(x_1,x_2)=f(x_1,x_2,\alpha_3,\ldots,\alpha_n)=x_1x_2+\alpha x_1+\beta x_2+\gamma,$$

где α, β, γ — константы, равные 0 или 1. Рассмотрим функцию $\psi(x_1,x_2)$, получаемую из $\varphi(x_1,x_2)$ следующим образом:

$$\psi(x_1,x_2)=\varphi(x_1+\beta,x_2+\alpha)+\alpha\beta+\gamma.$$

Очевидно, что

$$\varphi(x_1+\beta,x_2+\alpha)+\alpha\beta+\gamma=$$
$$=(x_1+\beta)(x_2+\alpha)+\alpha(x_1+\beta)+$$
$$+\beta(x_2+\alpha)+\gamma+\alpha\beta+\gamma=x_1x_2.$$

Следовательно,

$$\psi(x_1,x_2)=x_1\,\&\,x_2.$$

Лемма доказана полностью.

В заключение заметим, что замкнутые классы T_0, T_1, S, M и L попарно различны, что видно из табл. 8, в которой знак + означает, что функция содержится в классе, а знак — обозначает противоположную ситуацию.

Таблица 8

	T_0	T_1	S	M	L
0	+	—	—	+	+
1	—	+	—	+	+
x	—	—	+	—	+

Теперь мы можем перейти к рассмотрению одного из основных вопросов алгебры логики — вопроса о необхо-

димых и достаточных условиях полноты. Итак, пусть

$$\mathfrak{P} = \{f_1, f_2, \ldots, f_s, \ldots\}$$

— произвольная система функций из P_2. Спрашивается, как выяснить, будет ли она полной или нет? Ответ на этот вопрос дает следующая теорема.

Теорема 7 (о функциональной полноте). *Для того чтобы система функций $\mathfrak{P}$ была полной, необходимо и достаточно, чтобы она целиком не содержалась ни в одном из пяти замкнутых классов T_0, T_1, S, M и L.*

Доказательство. Необходимость. Пусть $\mathfrak{P}$ полна, т. е. $[\mathfrak{P}] = P_2$. Допустим, что $\mathfrak{P}$ содержится в одном из указанных классов — обозначим его через $\mathfrak{R}$, т. е. $\mathfrak{P} \subseteq \mathfrak{R}$. Тогда в силу свойств замыкания и замкнутости $\mathfrak{R}$ имеем

$$P_2 = [\mathfrak{P}] \subseteq [\mathfrak{R}] = \mathfrak{R}.$$

Значит $\mathfrak{R} = P_2$, что не так. Необходимость доказана.

Достаточность. Пусть $\mathfrak{P}$ целиком не содержится ни в одном из пяти указанных классов. Тогда из $\mathfrak{P}$ можно выделить подсистему $\mathfrak{P}'$, содержащую не более пяти функций, которая также обладает этим свойством. Для этого возьмем в $\mathfrak{P}$ функции f_i, f_j, f_k, f_m, f_l, которые не принадлежат соответственно классам T_0, T_1, S, M и L, и положим

$$\mathfrak{P}' = \{f_i, f_j, f_k, f_l, f_m\}.$$

Можно считать, что все эти функции зависят от одних и тех же переменных $x_1, \ldots, x_n$ (см. замечание из § 1).

Доказательство достаточности будем проводить в три этапа.

I. Построение при помощи функций f_i, f_j и f_k констант 0 и 1.

Рассмотрим функцию $f_i \notin T_0$. Возможны два случая:

1. $f_i(1, \ldots, 1) = 1$. Тогда $\varphi(x) = f_i(x, \ldots, x)$ есть константа 1, ибо

$$\varphi(0) = f_i(0, \ldots, 0) = 1, \quad \varphi(1) = f_i(1, \ldots, 1) = 1.$$

Вторая константа получается из f_j: $f_j(1, \ldots, 1) = 0$.

2. $f_i(1, \ldots, 1) = 0$. Тогда $\varphi(x) = f_i(x, \ldots, x)$ есть $\bar{x}$, ибо

$$\varphi(0) = f_i(0, \ldots, 0) = 1, \quad \varphi(1) = f_i(1, \ldots, 1) = 0.$$

Возьмем f_k $(f_k \notin S)$. Так как мы имеем $\bar{x}$, то в силу леммы 1 из f_k мы можем получить константу. Поскольку

мы располагаем $\bar{x}$, то находим и вторую константу. Итак, в обоих случаях мы получаем константы 0 и 1.

II. Построение при помощи констант 0, 1 и функции f_m функции $\bar{x}$. Это осуществляется на основе леммы 2.

III. Построение при помощи констант 0, 1 и функций $\bar{x}$ и f_l функции $x_1 \& x_2$. Это осуществляется на основе леммы 3.

Таким образом, мы при помощи формул над $\mathfrak{P}'$ (а значит и над $\mathfrak{P}$) реализовали функции $\bar{x}$ и $x_1 \& x_2$. Этим достаточность доказана.

Следствие 1. *Всякий замкнутый класс $\mathfrak{M}$ функций из P_2 такой, что $\mathfrak{M} \neq P_2$, содержится по крайней мере в одном из построенных классов.*

Определение. Класс $\mathfrak{N}$ функций из P_2 называется *предполным* (или *максимальным*), если $\mathfrak{N}$ неполный, а для любой функции $f (f \in P_2,\ f \notin \mathfrak{N})$ класс $\mathfrak{N} \cup \{f\}$ — полный.

Из определения следует, что предполный класс является замкнутым.

Следствие 2. *В алгебре логики существует только пять предполных классов, а именно: T_0, T_1, S, M и L.*

Рассмотрим пример, иллюстрирующий возможности теоремы о функциональной полноте.

Пример 15. Покажем, что система функций

$$f_1 = x_1 x_2,\ f_2 = 0,\ f_3 = 1 \text{ и } f_4 = x_1 + x_2 + x_3$$

является полной.

Мы имеем: $f_3 \notin T_0$, $f_2 \notin T_1$, $f_2 \notin S$, $f_4 \notin M$, $f_1 \notin L$. С другой стороны, удаление любой из функций приводит к неполной системе:

$$\{f_2, f_3, f_4\} \subset L, \quad \{f_1, f_3, f_4\} \subset T_1,$$

$$\{f_1, f_2, f_4\} \subset T_0, \quad \{f_1, f_2, f_3\} \subset M.$$

Из доказательства теоремы 7 непосредственно вытекает следующая теорема.

Теорема 8. *Из всякой полной в P_2 системы $\mathfrak{P}$ функций можно выделить полную подсистему, содержащую не более четырех функций.*

Доказательство. Мы видели, что из $\mathfrak{P}$ можно выделить полную подсистему $\mathfrak{P}'$, содержащую не более пяти функций. Оказывается, что функция $f_i \notin T_0$, кроме

того, либо не самодвойственна (случай 1), так как $f_i(0, \ldots, 0) = f_i(1, \ldots, 1)$, либо не сохраняет 1 и не монотонна (случай 2): $f_i(0, \ldots, 0) > f_i(1, \ldots, 1)$. Поэтому полной будет либо система $\{f_i, f_j, f_m, f_l\}$, либо система $\{f_i, f_k, f_l\}$.

Пример 15 показывает, что константа 4 не может быть понижена.

Теорема о функциональной полноте на самом деле дает не только критерий полноты. Она позволяет (в сочетании с разложением в д. н. ф. или к. н. ф.) найти для произвольной булевой функции f формулу через функции полной системы $\mathfrak{P}$.

§ 7. Представление о результатах Поста

Весьма глубокое изучение замкнутых классов в P_2 было осуществлено американским математиком Э. Постом. Им была описана структура всех замкнутых классов в P_2. Сформулируем некоторые из важнейших результатов, связанных с этим исследованием.

Определение. Система функций $\{f_1, f_2, \ldots, f_s, \ldots\}$ из замкнутого класса $\mathfrak{M}$ называется *полной в* $\mathfrak{M}$, если ее замыкание совпадает с $\mathfrak{M}$.

Иначе говоря, система полна в $\mathfrak{M}$, если каждая функция из $\mathfrak{M}$ может быть выражена в виде формулы через функции данной системы.

Определение. Система функций $\{f_1, f_2, \ldots, f_s, \ldots\}$ из замкнутого класса $\mathfrak{M}$ называется *его базисом*, если она полна в $\mathfrak{M}$, но всякая ее собственная подсистема не является полной в $\mathfrak{M}$.

Так, система

$$f_1 = x_1x_2, \quad f_2 = 0, \quad f_3 = 1, \quad f_4 = x_1 + x_2 + x_3$$

является базисом в P_2. Можно показать, что система функций $\{0, 1, x_1 \& x_2, x_1 \vee x_2\}$ является базисом для класса M.

Теорема 9 (Пост [47]). *Каждый замкнутый класс из P_2 имеет конечный базис.*

Теорема 10 (Пост [47]). *Мощность множества замкнутых классов в P_2 — счетная.*

Хотя вторая из этих теорем логически следует из первой, тем не менее в доказательстве Поста сначала устанавливается второй факт, а затем — первый.

Глава 2
k-ЗНАЧНАЯ ЛОГИКА

Конечнозначные логики вводятся как обобщение двузначной логики. В силу этого наше изложение местами будет кратким, а некоторые аналогичные определения и доказательства будут опущены. Особое внимание обратим на два обстоятельства:

1) в k-значных логиках сохраняются многие свойства и результаты, которые имели место в двузначной логике;

2) в k-значных логиках наблюдаются явления, обнаруживающие принципиальное их отличие от алгебры логики.

В связи с этим некоторые задачи не имеют такого исчерпывающего решения как в алгебре логики, а другие вовсе не решены.

§ 1. Функции k-значной логики. Формулы и реализация функций формулами

Пусть $U = \{u_1, u_2, \ldots, u_m, \ldots\}$ — исходный алфавит переменных (аргументов). Будем рассматривать функции $f(u_{i_1}, \ldots, u_{i_n})$ $(u_{i_\nu} \neq u_{i_\mu}$ при $\nu \neq \mu)$, аргументы которых определены на множестве $E_k = \{0, 1, \ldots, k-1\}$, и такие, что $f(\alpha_1, \ldots, \alpha_n) \in E_k$, когда $\alpha_i \in E_k$ $(i = 1, 2, \ldots, n)$.

Для упрощения записи мы будем использовать для переменных из U метаобозначения $x, y, z, \ldots$, а также $x_i, y_i, z_i, \ldots$ и употреблять для функций более простую запись $f(x_1, \ldots, x_n)$.

Очевидно, что функция $f(x_1, \ldots, x_n)$ полностью определена, если задана ее таблица (см. табл. 1). В этой таблице наборы суть разложения в k-ичной системе счисления чисел $0, 1, \ldots, k^n - 1$. Символ f здесь будет интерпретироваться как символ, обозначающий отображение, характеризуемое таблицей, а символы $x_1, x_2, \ldots, x_n$ — как названия столбцов. Для функций одной переменной мы наряду с таблицами будем использовать запись в виде (обобщенной) подстановки

$$S(x) = \begin{pmatrix} 0 & 1 \ldots k-1 \\ i_0 & i_1 \ldots i_{k-1} \end{pmatrix},$$

где $S(\alpha) = i_\alpha$.

Обозначим через P_k множество всех функций k-значной логики над алфавитом U, а также констант 0, 1, ..., $k-1$. Так как число наборов $(\alpha_1, \ldots, \alpha_n)$ значений переменных $x_1, \ldots, x_n$ равно k^n, то имеем следующий результат.

Таблица 1

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

Теорема 1. *Число всех функций из P_k, зависящих от n переменных $x_1, \ldots, x_n$, равно k^{k^n}.*

Из сказанного вытекает, что в P_k при $k \geqslant 3$ в значительной степени возрастают трудности по сравнению с P_2 как в возможности эффективного использования табличного задания функций, так и в возможности просмотра всех функций от n переменных. Уже в P_3 число функций от двух переменных равно $3^9 = 19683$, т. е. это множество практически необозримо. В P_k часто употребляют вместо табличного задания функций задание при помощи алгоритма вычислимости функций. Например,

$$\max(x_1, \ldots, x_n)$$

можно рассматривать, как алгоритм, который для любого набора $(\alpha_1, \ldots, \alpha_n)$ значений переменных выдает их максимум. Этот алгоритм определяет в P_k единственную функцию, которую мы будем обозначать тем же символом.

Далее вводится (как в P_2) понятие существенной и несущественной переменных, а также понятие равенства функций. Это позволяет рассматривать функции в P_k с точностью до фиктивных переменных.

После этого рассматриваются примеры некоторых конкретных функций из P_k, которые можно считать «элементарными» функциями.

1) $\bar{x} = x + 1 \pmod k$. Здесь $\bar{x}$ представляет обобщение отрицания в смысле «циклического» сдвига значений.

2) $Nx = k - 1 - x$. Здесь Nx или, как часто обозначают, $\sim x$ является другим обобщением отрицания в смысле «зеркального» отображения значений. В литературе оно носит название *отрицания Лукашевича.*

3) $I_i(x) = \begin{cases} k-1 & \text{при } x = i, \\ 0 & \text{при } x \neq i \end{cases}$ $(i = 0, \ldots, k-1)$.

Функции $I_i(x)$ при $i \neq k-1$ являются обобщениями некоторых свойств отрицания.

4) $j_i(x) = \begin{cases} 1 & \text{при } x = i, \\ 0 & \text{при } x \neq i. \end{cases}$

Функция $j_i(x)$ — характеристическая функция значения i и при $i \neq k-1$ представляет собой обобщение отрицания.

5) $\min(x_1, x_2)$ — обобщение конъюнкции.

6) $x_1 x_2 \pmod k$ — второе обобщение конъюнкции.

7) $\max(x_1, x_2)$ — обобщение дизъюнкции.

8) $x_1 + x_2 \pmod k$.

Из рассмотрения этого списка элементарных функций видно, что функции алгебры логики имеют в k-значной логике ($k \geqslant 3$) по нескольку аналогов, каждый из которых обобщает соответствующее свойство функции.

Затем, так же как и в алгебре логики, вводится понятие формулы над множеством функций $\mathfrak{P}$. Формулы мы обозначаем символами $\mathfrak{A}$, $\mathfrak{B}$,... без индексов и с индексами. Если мы хотим указать зависимость формулы от переменных или выделить функции, из которых построена формула, то употребляем обозначения

$$\mathfrak{A}(x_1, \ldots, x_n), \quad \mathfrak{A}[f_1, \ldots, f_s].$$

Каждой формуле $\mathfrak{A}(x_1, \ldots, x_n)$ сопоставляется функция $f(x_1, \ldots, x_n)$ из P_k, при этом говорят также, что формула $\mathfrak{A}$ реализует функцию f. Аналогичный смысл здесь имеют суперпозиция функций из $\mathfrak{P}$ и операция суперпозиции. Далее вводится понятие эквивалентности формул $\mathfrak{A}$ и $\mathfrak{B}$: $\mathfrak{A} = \mathfrak{B}$, если соответствующие им функции $f_{\mathfrak{A}}$ и $f_{\mathfrak{B}}$ равны.

Опираясь на понятие эквивалентности, можно описать основные свойства элементарных функций. Укажем некоторые из них. Пусть $(x_1 \circ x_2)^*$ обозначает любую из функций $\min(x_1, x_2)$, $x_1 x_2 \pmod k$, $\max(x_1, x_2)$, $x_1 + x_2 \pmod k$.

1. Функция $(x_1 \circ x_2)$ обладает свойством ассоциативности:

$$((x_1 \circ x_2) \circ x_3) = (x_1 \circ (x_2 \circ x_3)).$$

2. Функция $(x_1 \circ x_2)$ обладает свойством коммутативности:

$$(x_1 \circ x_2) = (x_2 \circ x_1).$$

Далее мы иногда будем обозначать функции $\min(x_1, x_2)$ и $\max(x_1, x_2)$ соответственно через $(x_1 \& x_2)$ и $(x_1 \vee x_2)$. В силу свойства ассоциативности и соглашения о том, что операция $\&$ выполняется раньше операции $\vee$ (см. замечания 1—3 § 3 гл. 1), можно употреблять для записи формул выражения, получающиеся из формул путем опускания некоторых скобок.

Укажем еще несколько групп тождеств (правил), относящихся уже к системе элементарных функций

$$\{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}.$$

3. Правила спуска символа I «вглубь» формулы:

$$I_\sigma(c) = \begin{cases} k-1 & \text{при } c = \sigma, \\ 0 & \text{при } c \neq \sigma \end{cases} \quad (\sigma, c = 0, 1, \ldots, k-1);$$

$$I_\sigma(I_\tau(x)) = \begin{cases} I_0(x) \vee \ldots \vee I_{\tau-1}(x) \vee I_{\tau+1}(x) \vee \ldots \vee I_{k-1}(x) & \\ & \text{при } \sigma = 0, \\ 0 \quad \text{при} \quad 0 < \sigma < k-1, & \\ I_\tau(x) \quad \text{при} \quad \sigma = k-1; & \end{cases}$$

$$I_\sigma(x_1 x_2) = I_\sigma(x_1)\,(I_\sigma(x_2) \vee \ldots \vee I_{k-1}(x_2)) \vee \vee I_\sigma(x_2)\,(I_\sigma(x_1) \vee \ldots \vee I_{k-1}(x_1));$$

$$I_\sigma(x_1 \vee x_2) = I_\sigma(x_1)\,(I_0(x_2) \vee \ldots \vee I_\sigma(x_2)) \vee \vee I_\sigma(x_2)\,(I_0(x_1) \vee \ldots \vee I_\sigma(x_1)).$$

4. Свойства дистрибутивности:

$$(x_1 \vee x_2) x_3 = (x_1 x_3) \vee (x_2 x_3),$$

$$(x_1 x_2) \vee x_3 = (x_1 \vee x_3)(x_2 \vee x_3).$$

5. Правило исключения «чистых» вхождений переменной:

$$x = 1 \cdot I_1(x) \vee 2 \cdot I_2(x) \vee \ldots \vee (k-1) I_{k-1}(x).$$

6. Правило введения переменной:

$$x_1 = x_1(I_0(x_2) \vee \ldots \vee I_{k-1}(x_2)).$$

7. Правила упрощений:

$$I_\sigma(x)\, I_\tau(x) = \begin{cases} I_\sigma(x) & \text{при } \tau = \sigma, \\ 0 & \text{при } \tau \neq \sigma; \end{cases}$$

$$\sigma\tau = \min(\sigma, \tau); \quad \sigma \vee \tau = \max(\sigma, \tau);$$

$$(k-1)x = x; \quad 0 \cdot x = 0;$$

$$(k-1) \vee x = k-1; \quad 0 \vee x = x.$$

Опираясь на некоторый запас тождеств (например, тождества 1—7) для системы элементарных функций

$$\{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\},$$

можно при помощи эквивалентных преобразований получить новые тождества.

Рассмотрение свойств элементарных функций показывает, что не для всех обобщений булевых функций сохраняются соответствующие свойства. Поясним это на примерах.

Пример 1.

1) $\sim(\sim x) = x$, но $\overline{\overline{x}} \neq x$ (при $k \geqslant 3$).
2) $\sim\min(x_1, x_2) = \max(\sim x_1, \sim x_2)$, но $\overline{\min(x_1, x_2)} \neq \max(\bar{x}_1, \bar{x}_2)$.

В заключение приведем тождество, представляющее аналог совершенной д. н. ф. и играющее важную роль в последующем изложении:

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_n)} I_{\sigma_1}(x_1) \& \ldots \& I_{\sigma_n}(x_n) \,\&\, f(\sigma_1, \ldots, \sigma_n).$$

Доказывается оно прямой проверкой.

Нетрудно видеть, что каждая формула над множеством элементарных функций $\{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$ может быть при помощи тождеств 1—7 преобразована к д. н. ф. Отсюда легко показать, что правила 1—7 позволяют перейти от любой формулы над данным множеством к любой другой формуле, эквивалентной исходной. Последнее свидетельствует о том, что система тождеств 1—7 в известном смысле обладает полнотой.

§ 2. Примеры полных систем

В P_k определение полноты выглядит так же, как и в P_2.

Определение. Система $\mathfrak{P}$ функций $f_1, f_2, \ldots, f_s, \ldots$ из P_k называется (функционально) *полной*, если любая функция из P_k может быть записана в виде формулы через функции этой системы.

Ниже рассматриваются примеры полных систем. Для обоснования полноты мы будем использовать принцип сведения задачи о полноте одних систем к задаче о полноте других (см. § 5, гл. 1).

1. Система $\mathfrak{P} = P_k$ полна. Очевидно, что множество всех функций из P_k представляет полную систему.

2. Система

$$\mathfrak{P} = \{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$$

является полной.

Теорема 2. *Система*

$$\mathfrak{P} = \{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$$

является полной в P_k.

Доказательство. Пусть $f(x_1, \ldots, x_n)$ — произвольная функция из P_k. Для нее имеет место разложение

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_n)} I_{\sigma_1}(x_1) \,\&\, \ldots \,\&\, I_{\sigma_n}(x_n) \,\&\, f(\sigma_1, \ldots, \sigma_n).$$

Данная формула (правая часть) построена из функций, входящих в $\mathfrak{P}$. Теорема доказана.

3. Система $\mathfrak{P} = \{\bar{x}, \max(x_1, x_2)\}$ является полной.

Теорема 3. *Система* $\mathfrak{P} = \{\bar{x}, \max(x_1, x_2)\}$ *полна в* P_k.

Доказательство. а) Построение констант. Из функции $\bar{x} = x + 1$ получаем функции

$$x + 2 = (x + 1) + 1, \ldots, x + k - 1 = (x + k - 2) + 1,$$
$$x = x + k = (x + (k - 1)) + 1.$$

Легко видеть, что

$$\max(x, x + 1, \ldots, x + k - 1) = k - 1.$$

Отсюда при помощи $\bar{x}$ получаем остальные константы

б) Построение функций одной переменной. Сначала получаем функции $I_i(x)$ $(i=0,\ldots,k-1)$:

$$I_i(x)=1+\max_{\alpha\neq k-1-i}\{x+\alpha\}.$$

В самом деле, если $x=i$, то левая часть равна $k-1$, а правая часть есть

$$1+\max_{\alpha\neq k-1-i}\{i+\alpha\}=1+\max_{i+\alpha\neq k-1}\{i+\alpha\}=$$

$$=1+k-2=k-1;$$

если $x\neq i$, то левая часть равна 0, а правая —

$$1+\max_{\alpha\neq k-1-i}\{x+\alpha\}=1+(x+(k-1-x))=k=0.$$

Введем функции $f_{s,i}(x)$, где

$$f_{s,i}(x)=\begin{cases}s & \text{при } x=i,\\ 0 & \text{при } x\neq i.\end{cases}$$

Покажем, что

$$f_{s,i}(x)=s+1+\max[I_i(x),\ k-1-s].$$

Последнее легко усматривается из графиков, изображенных на рис. 1.

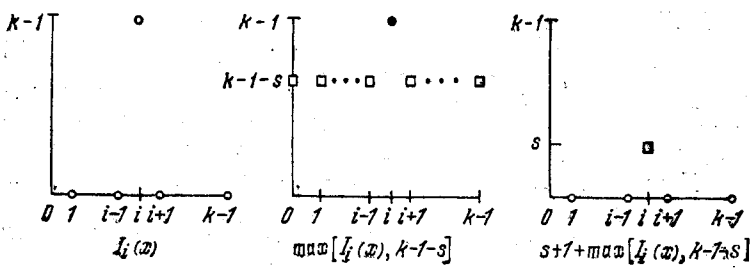


Рис. 1

Если $g(x)$ — произвольная функция одной переменной из P_k, то

$$g(x)=\max\{f_{g(0),0}(x),\ f_{g(1),1}(x),\ \ldots,\ f_{g(k-1),k-1}(x)\},$$

и, в частности, $\sim x=\max\{f_{k-1,0}(x),\ f_{k-2,1}(x),\ldots,f_{0,k-1}(x)\}$.

в) Получение $\min(x_1, x_2)$. Мы видели, что

$$\sim\min(x_1,\ x_2)=\max(\sim x_1,\ \sim x_2).$$

Отсюда

$$\min(x_1, x_2) = \sim\max(\sim x_1, \sim x_2).$$

Таким образом, мы можем при помощи формул над исходной системой функций выразить любую из функций системы

$$\{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\},$$

относительно которой доказано, что она полная. Поэтому и система $\{\bar{x}, \max(x_1, x_2)\}$ является полной. Теорема доказана.

Введем обозначение: $V_k(x_1, x_2) = \max(x_1, x_2) + 1$. Функция $V_k(x_1, x_2)$ называется *функцией Вебба* и представляет собой аналог функции Шеффера.

4. Система $\mathfrak{P} = \{V_k(x_1, x_2)\}$ является полной. Вопрос о ее полноте может быть легко сведен к полноте системы 3.

С понятием полноты связано понятие замыкания и замкнутого класса.

Определение. Пусть $\mathfrak{M}$ — произвольное подмножество функций из P_k. *Замыканием* $\mathfrak{M}$ называется множество $[\mathfrak{M}]$ всех функций из P_k, представимых в виде формул через функции множества $\mathfrak{M}$.

Определение. Класс (множество) $\mathfrak{M}$ называется (функционально) *замкнутым*, если $[\mathfrak{M}] = \mathfrak{M}$.

Для данных понятий справедливы высказывания, которые были сделаны при рассмотрении аналогичных понятий в P_2. Приведем примеры замкнутых множеств.

Пример 2. 1) Класс $\mathfrak{M} = P_k$, очевидно, является замкнутым.

2) Пусть $\mathscr{E} \subset E_k$. Обозначим через $T_{\mathscr{E}}$ множество всех функций $f(x_1, \ldots, x_n)$ из P_k таких, что $f(\alpha_1, \ldots, \alpha_n) \in \mathscr{E}$, если $\alpha_i \in \mathscr{E}$ $(i = 1, \ldots, n)$. Другими словами, $T_{\mathscr{E}}$ — множество всех функций из P_k, *сохраняющих* $\mathscr{E}$; будем записывать это так:

$$f(\mathscr{E}, \mathscr{E}, \ldots, \mathscr{E}) \subseteq \mathscr{E}.$$

Класс $\mathfrak{M} = T_{\mathscr{E}}$, очевидно, является замкнутым.

В терминах замыкания можно дать другое определение полноты, эквивалентное исходному: $\mathfrak{M}$ — полная система, если $[\mathfrak{M}] = P_k$.

Понятие замкнутого класса может быть приложено к решению вопроса об обосновании неполноты некоторых систем.

Пример 3. Рассмотрим систему $\mathfrak{P}=\{\sim x, \max(x_1, x_2)\}$. Пусть $\mathscr{E} = \{0, k-1\}$. Так как обе функции сохраняют $\mathscr{E}$, то

$$[\mathfrak{P}] \subseteq T_{\mathscr{E}}.$$

Поскольку при $k \geqslant 3$, $\mathscr{E} \neq E_k$, то $T_{\mathscr{E}}$ не содержит, например, константу 1. Значит при $k \geqslant 3$ $\mathfrak{P}$ не будет полной системой. На этом примере видно, что, хотя система $\{\sim x, \max(x_1, x_2)\}$ и является обобщением системы $\{\bar{x}, x_1 \vee x_2\}$ булевых функций, она не является полной.

§ 3. Распознавание полноты. Теорема о полноте

Теперь мы перейдем к обсуждению вопросов полноты для произвольных систем $\mathfrak{P}$. Следовательно, здесь нас будет интересовать, каким образом по множеству $\mathfrak{P}$ можно узнать, будет оно полным или нет. Мы рассмотрим два подхода к решению этой задачи.

Первый подход алгоритмический. Он требует уточнения слов «задана произвольная система $\mathfrak{P}$». Ввиду этого мы несколько сузим задачу и будем предполагать, что система $\mathfrak{P}$ конечна:

$$\mathfrak{P} = \{f_1, f_2, \ldots, f_s\}$$

и, стало быть, она может быть явно задана либо перечнем таблиц, либо списком формул. Так же как и в случае $k = 2$, можно считать, что каждая из функций системы $\mathfrak{P}$ зависит от переменных

$$x_1, x_2, \ldots, x_n.$$

Наша задача может быть сформулирована следующим образом: существует ли алгоритм, позволяющий для каждой конечной системы $\mathfrak{P}$ выяснять, будет она полной или нет (*задача распознавания полноты*).

Для произвольного $p \geqslant 1$ обозначим

$$g_i^p(x_1, \ldots, x_p) = x_i$$

и через $\mathfrak{M}_{x_1 \ldots x_p}$ — множество всех функций из $\mathfrak{M}$, зависящих от переменных $x_1, \ldots, x_p$.

Теорема 4. *Существует алгоритм для распознавания полноты.*

Доказательство. Построим по индукции последовательность множеств

$$\mathfrak{N}_0, \mathfrak{N}_1, \ldots, \mathfrak{N}_r, \ldots$$

функций от двух переменных x_1 и x_2.

Базис индукции. Положим $\mathfrak{N}_0 = \Lambda$, где Λ — пустое множество.

Индуктивный переход. Пусть уже построены множества $\mathfrak{N}_0, \mathfrak{N}_1, \ldots, \mathfrak{N}_r$; покажем, как определяется множество $\mathfrak{N}_{r+1}$. Для этого выпишем функции, входящие в $\mathfrak{N}_r$ $(r \geqslant 0)$:

$$\mathfrak{N}_r = \{h_1(x_1, x_2), \ldots, h_{s_r}(x_1, x_2)\} \quad (s_r = 0 \text{ при } r = 0),$$

и для каждого i $(i = 1, \ldots, s)$ рассмотрим всевозможные формулы вида

$$f_i(H_1(x_1, x_2), \ldots, H_n(x_1, x_2)),$$

где $H_l(x_1, x_2)$ есть либо функция $h_j(x_1, x_2)$ $(j = 1, \ldots, s_r)$, либо $g_\sigma^2(x_1, x_2)$.

Таким образом, просматривая $s(s_r + 2)^n$ формул, мы, быть может, получим функции, не вошедшие в $\mathfrak{N}_r$. Обозначим их через

$$h_{s_r+1}(x_1, x_2), \ldots, h_{s_{r+1}}(x_1, x_2).$$

Положим $\mathfrak{N}_{r+1} = \mathfrak{N}_r \cup \{h_{s_r+1}(x_1, x_2), \ldots, h_{s_{r+1}}(x_1, x_2)\}$. Очевидно, что

$$\mathfrak{N}_0 \subseteq \mathfrak{N}_1 \subseteq \ldots \subseteq \mathfrak{N}_r \subseteq \ldots$$

Из построения ясно, что если $\mathfrak{N}_{r+1} = \mathfrak{N}_r$, то $\mathfrak{N}_r = \mathfrak{N}_{r+1} = \ldots$, т. е. цепочка множеств стабилизируется. Обозначим через r^* минимальный номер множества, начиная с которого наступает стабилизация. Тогда цепочка множеств

$$\mathfrak{N}_0 \subset \mathfrak{N}_1 \subset \ldots \subset \mathfrak{N}_{r^*}$$

строго возрастает. Так как мощность $\mathfrak{N}_i$ не больше, чем k^{k^2}, то $r^* \leqslant k^{k^2}$. Значит, момент стабилизации может быть обнаружен через ограниченное число шагов. Рассмотрим множество $\mathfrak{N}_{r^*}$. Возможны два случая.

1) $\mathfrak{N}_{r^*}$ содержит все функции от двух переменных x_1, x_2 и, значит, содержит $V_k(x_1, x_2)$. Тогда исходная система полна.

2) $\mathfrak{N}_{r^*}$ не содержит всех функций от двух переменных. Поскольку в этом случае $[\mathfrak{P}]_{x_1x_2} = \mathfrak{N}_{r^*}$ (см. заме-

чание 1 из § 2 гл. 1), то $[\mathfrak{P}]$ не содержит всех функций от переменных x_1 и x_2. Следовательно, $\mathfrak{P}$ не полна.

Из данных рассуждений легко извлекается алгоритм: строим классы $\mathfrak{R}_0, \mathfrak{R}_1, \ldots, \mathfrak{R}_{r*}$ до момента стабилизации и рассматриваем класс $\mathfrak{R}_{r*}$, по которому и определяем, имеет место полнота для $\mathfrak{P}$ или нет. Теорема доказана.

Теорема 5. *Из всякой полной в P_k системы $\mathfrak{P}$ можно выделить конечную подсистему, являющуюся также полной.*

Доказательство. Пусть $\mathfrak{P} = \{f_1, f_2, \ldots, f_s, \ldots\}$. В силу полноты функция $V_k(x_1, x_2)$ может быть выражена через функции системы $\mathfrak{P}$ в виде формулы

$$V_k(x_1, x_2) = \mathfrak{U}[f_{i_1}, \ldots, f_{i_r}].$$

Очевидно, что подсистема $\{f_{i_1}, \ldots, f_{i_r}\}$ является искомой. Теорема доказана.

Таким образом, существенно бесконечных полных систем не бывает, и тем самым введенное выше ограничение в задаче распознавания полноты является не столь сильным, как это могло казаться первоначально.

Второй подход в решении вопроса о полноте связан с проверкой некоторых свойств класса $\mathfrak{P}$.

Введем одно понятие и докажем относительно него две леммы.

Пусть $\mathfrak{R}$ — класс функций $h_j(y_1, \ldots, y_p)$ из P_k, зависящих от p переменных $y_1, \ldots, y_p$. Предположим, что он содержит функции $g_i(y_1, \ldots, y_p) = y_i$ $(i = 1, \ldots, p)$.

Определение. Функция $f(x_1, \ldots, x_n)$ *сохраняет множество* $\mathfrak{R}$, если для любых функций $h_{i_1}(y_1, \ldots, y_p), \ldots, h_{i_n}(y_1, \ldots, y_p)$ из $\mathfrak{R}$

$$f(h_{i_1}(y_1, \ldots, y_p), \ldots, h_{i_n}(y_1, \ldots, y_p)) \in \mathfrak{R}.$$

Обозначим через $\mathfrak{M}$ класс всех функций из P_k, сохраняющих множество $\mathfrak{R}$.

Пример 4. Пусть $k = 2$, $p = 1$ и $\mathfrak{R} = \{y, \bar{y}\}$. Тогда функция $f(x_1, \ldots, x_n)$, сохраняющая множество $\mathfrak{R}$, удовлетворяет соотношению

$$f(y^{\sigma_1}, \ldots, y^{\sigma_n}) = y^{\sigma},$$

откуда

$$(\bar{\sigma}_1, \ldots, \bar{\sigma}_n) = \bar{f}(\sigma_1, \ldots, \sigma_n).$$

Значит, функция $f(x_1, \ldots, x_n)$ — самодвойственная, и класс

сохранения множества $\mathfrak{N}$ есть класс S самодвойственных функций.

Лемма 1. *Класс $\mathfrak{M}$ всех функций, сохраняющих $\mathfrak{N}$, является замкнутым.*

Доказательство. Очевидно, что класс $\mathfrak{M}$ содержит тождественную функцию. Поэтому для обоснования замкнутости класса $\mathfrak{M}$ достаточно показать, что функция $\Phi = f(f_1, \ldots, f_m)$ принадлежит классу $\mathfrak{M}$, если функции $f, f_1, \ldots, f_m$ принадлежат классу $\mathfrak{M}$. Пусть Φ зависит от n переменных. Возьмем произвольные функции $h_{i_1}, \ldots, h_{i_n}$ из класса $\mathfrak{N}$. Тогда

$$\Phi(h_{i_1}, \ldots, h_{i_n}) = f(f_1(h_{i_1}, \ldots, h_{i_n}), \ldots, f_m(h_{i_1}, \ldots, h_{i_n})) = \\ = f(H_1, \ldots, H_m),$$

где функции $H_1, \ldots, H_m$ принадлежат классу $\mathfrak{N}$, поэтому и $f(H_1, \ldots, H_m)$ принадлежит $\mathfrak{N}$. Лемма доказана.

Лемма 2. *Если класс $\mathfrak{N}$ таков, что $[\mathfrak{N}]_{y_1 \ldots y_p} = \mathfrak{N}$, то для класса $\mathfrak{M}$, сохраняющего $\mathfrak{N}$, имеет место равенство*

$$\mathfrak{M}_{y_1 \ldots y_p} = \mathfrak{N}.$$

Доказательство. Пусть $h(y_1, \ldots, y_p) \in \mathfrak{N}$. Тогда если $h_{i_1}, \ldots, h_{i_p} \in \mathfrak{N}$, то

$$h(h_{i_1}, \ldots, h_{i_p}) \in [\mathfrak{N}]_{y_1 \ldots y_p} = \mathfrak{N},$$

т. е. $h \in \mathfrak{M}_{y_1 \ldots y_p}$. С другой стороны, если

$$f(y_1, \ldots, y_p) \in \mathfrak{M}_{y_1 \ldots y_p},$$

то, подставляя вместо $y_1, \ldots, y_p$ функции $g_1, \ldots, g_p$, получим

$$f(g_1, \ldots, g_p) \in \mathfrak{N} \quad \text{или} \quad f(y_1, \ldots, y_p) \in \mathfrak{N}.$$

Лемма доказана.

Теорема 6 (о функциональной полноте, А. В. Кузнецов [15]). *Можно построить систему замкнутых классов в P_k*

$$\mathfrak{M}_1, \mathfrak{M}_2, \ldots, \mathfrak{M}_s,$$

каждый из которых целиком не содержит ни одного из остальных классов, и такую, что подсистема функций из P_k полна тогда и только тогда, когда она целиком не содержится ни в одном из классов $\mathfrak{M}_1, \ldots, \mathfrak{M}_s$.

Доказательство. Построение системы классов $\mathfrak{M}_1, \ldots, \mathfrak{M}_s$. Пусть $\mathfrak{N}_1, \mathfrak{N}_2, \ldots, \mathfrak{N}_l$ — система всех таких собственных подмножеств функций из P_k, зависящих от двух переменных x_1 и x_2, которые удовлетворяют следующим условиям ($i = 1, \ldots, l$):

1) $\mathfrak{N}_i$ содержит обе функции $g_1(x_1, x_2) = x_1$, $g_2(x_1, x_2) = x_2$;

2) $[\mathfrak{N}_i]_{x_1x_2} = \mathfrak{N}_i$.

Указанные подмножества строятся путем просмотра всех собственных подмножеств множества функций из P_k, зависящих от переменных x_1 и x_2 $\left(\text{их} < 2^{k^{k^2}}\right)$. При этом оставляются те подмножества, которые содержат обе функции g_1 и g_2, и далее для каждого оставшегося подмножества проверяют условие $[\mathfrak{N}]_{x_1x_2} = \mathfrak{N}_i$, что может быть осуществлено так же, как в теореме о распознавании полноты.

Обозначим далее через $\mathfrak{M}'_i$ класс сохранения подмножества $\mathfrak{N}_i$. В силу лемм 1 и 2 $\mathfrak{M}'_i$ — замкнутый класс такой, что

$$(\mathfrak{M}'_i)_{x_1x_2} = \mathfrak{N}_i.$$

Отсюда следует, что все классы $\mathfrak{M}'_i$ ($i = 1, \ldots, l$) различны и не являются полными.

Далее остается только удалить те классы, которые содержатся в каком-либо из остальных классов. Мы получим систему

$$\mathfrak{M}_1, \mathfrak{M}_2, \ldots, \mathfrak{M}_s.$$

Необходимость вытекает из свойств замкнутости и неполноты классов $\mathfrak{M}_j$ ($j = 1, \ldots, s$).

Достаточность. Пусть $\mathfrak{M} \subseteq P_k$ — система функций, целиком не содержащаяся ни в одном из классов $\mathfrak{M}_j$ ($j = 1, \ldots, s$). Можно считать, что $\mathfrak{M}$ — замкнутый класс. Обозначим через $\mathfrak{M}'$ класс $[\mathfrak{M} \cup \{g_1, g_2\}]$. Очевидно, что классы $\mathfrak{M}$ и $\mathfrak{M}'$ либо одновременно полны, либо неполны, так как

$$\mathfrak{M}' = \mathfrak{M} \cup [\{g_1, g_2\}],$$

и функция $V_k(x_1, x_2)$ либо входит в $\mathfrak{M}$ и $\mathfrak{M}'$, либо не содержится ни в одном из этих классов. Возьмем $\mathfrak{N}' = \mathfrak{M}'_{x_1x_2}$. Покажем, что $\mathfrak{N}'$ содержит все функции, зависящие от

переменных x_1 и x_2. В самом деле, если это не так, то очевидно, что

$$\mathfrak{N}' \equiv \mathfrak{N}_i \quad \text{и} \quad \mathfrak{M}' \subseteq \mathfrak{M}_i' \subseteq \mathfrak{M}_j.$$

Так как $\mathfrak{M} \subseteq \mathfrak{M}'$, то получаем, что $\mathfrak{M} \subseteq \mathfrak{M}_j$, что противоречиво. Таким образом, $\mathfrak{N}'$, а значит и $\mathfrak{M}'$, содержит функцию $V_k(x_1, x_2)$. Отсюда вытекает полнота класса $\mathfrak{M}'$, а следовательно и класса $\mathfrak{M}$. Теорема доказана.

Следует обратить внимание на то обстоятельство, что теорема Кузнецова доказывает, что возможно выразить условия полноты системы $\mathfrak{P}$ в терминах принадлежности ее к специальным классам $\mathfrak{M}_1, \ldots, \mathfrak{M}_s$. Однако фактическое построение классов даже при небольших k связано с трудоемкими вычислениями, которые невозможно осуществить. В силу этого возникает вопрос о поиске других, более эффективных критериев. Мы увидим, что эта цель достижима, но за счет введения ограничений, т. е. за счет знания дополнительной информации об исходной системе $\mathfrak{P}$.

§ 4. Некоторые свойства существенных функций. Критерий полноты

Целью этого параграфа является доказательство одного критерия полноты. Однако для этого нам необходимо несколько подробнее изучить свойства функций $f(x_1, \ldots, x_n)$ из P_k, которые зависят существенно не менее чем от двух переменных. Данные функции мы будем называть *существенными*. Докажем три леммы.

Лемма 3. *Пусть $f(x_1, \ldots, x_n)$ — существенная функция, принимающая l ($l \geqslant 3$) значений. Пусть x_1 — ее существенная переменная. Тогда найдутся два таких набора $(\alpha, \alpha_2, \ldots, \alpha_n)$ и $(\beta, \alpha_2, \ldots, \alpha_n)$, что*

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\beta, \alpha_2, \ldots, \alpha_n)$$

и $f(\alpha, x_2, \ldots, x_n)$ принимает значение, отличное и от $f(\alpha, \alpha_2, \ldots, \alpha_n)$, и от $f(\beta, \alpha_2, \ldots, \alpha_n)$.

Доказательство. Так как x_1 — существенная переменная функции f, то найдутся такие значения $\alpha_2, \ldots, \alpha_n$, что последовательность

$$f(0, \alpha_2, \ldots, \alpha_n), f(1, \alpha_2, \ldots, \alpha_n), \ldots$$
$$\ldots, f(k-1, \alpha_2, \ldots, \alpha_n) \qquad (*)$$

содержит не менее двух различных значений. Возможны два случая.

1) В последовательности (*) содержатся не все l значений. Рассмотрим набор $(\alpha, \gamma_2, \ldots, \gamma_n)$ такой, что значение $f(\alpha, \gamma_2, \ldots, \gamma_n)$ не встречается в (*). Очевидно, что

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

Примем за β любое из значений, для которых

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \alpha_2, \ldots, \alpha_n).$$

Очевидно также, что

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

2) В последовательности (*) встречаются все l значений. Из существенности функции f вытекает, что существует α такое, что

$$f(\alpha, x_2, \ldots, x_n) \neq \text{const}$$

(иначе бы f зависела существенно только от одной переменной). Отсюда следует, что найдутся наборы $(\alpha, \alpha_2, \ldots, \alpha_n)$ и $(\alpha, \gamma_2, \ldots, \gamma_n)$ такие, что

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

Так как последовательность (*) содержит l ($l \geqslant 3$) значений, то найдется такое β, что

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \alpha_2, \ldots, \alpha_n),$$

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

Лемма доказана.

Пусть G — конечное множество. Обозначим через $|G|$ число элементов в G.

Лемма 4 (основная). *Если $f(x_1, \ldots, x_n)$ — существенная функция, принимающая не менее l ($l \geqslant 3$) значений, то найдутся n подмножеств $G_1, \ldots, G_n$ множества E_k таких, что*

$$1 \leqslant |G_1|, \ldots, |G_n| \leqslant l - 1$$

и на множестве наборов $(\alpha_1, \ldots, \alpha_n)$, где $\alpha_i \in G_i$ ($i = 1, \ldots, n$), т. е. на $G_1 \times G_2 \times \ldots \times G_n$, функция f принимает l значений.

Доказательство. Можно считать, что x_1 — существенная переменная функции f. На основании предыду-

щей леммы найдутся три набора

$$(\alpha, \alpha_2, \ldots, \alpha_n), \quad (\beta, \alpha_2, \ldots, \alpha_n), \quad (\alpha, \gamma_2, \ldots, \gamma_n),$$

на которых f принимает три различных значения. Добавим к этим наборам еще $l-3$ набора

$$(\delta_1^{(1)}, \delta_2^{(1)}, \ldots, \delta_n^{(1)}), \ldots, (\delta_1^{(l-3)}, \delta_2^{(l-3)}, \ldots, \delta_n^{(l-3)}),$$

на которых f принимает остальные $l-3$ значений.

Положим

$$G_1 = \{\alpha, \beta, \delta_1^{(1)}, \ldots, \delta_1^{(l-3)}\},$$
$$G_2 = \{\alpha_2, \gamma_2, \delta_2^{(1)}, \ldots, \delta_2^{(l-3)}\},$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$G_n = \{\alpha_n, \gamma_n, \delta_n^{(1)}, \ldots, \delta_n^{(l-3)}\}.$$

Построенные множества, очевидно, удовлетворяют условиям леммы. Лемма доказана.

Определение. Система наборов вида

$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n)$$

называется *квадратом*, если $\alpha_i \neq \beta_i$ и $\alpha_j \neq \beta_j$.

Лемма 5. *Пусть $f(x_1, \ldots, x_n)$ — существенная функция, принимающая l $(l \geqslant 3)$ значений. Тогда найдется квадрат, на котором f принимает либо более двух значений, либо два значения, причем одно из них только в одной его точке.*

Доказательство. Можно считать, что x_1 — существенная переменная функции f. Тогда на основании леммы 3 найдутся наборы

$$(\alpha, \alpha_2, \ldots, \alpha_n), \quad (\beta, \alpha_2, \ldots, \alpha_n), \quad (\alpha, \gamma_2, \ldots, \gamma_n),$$

на которых f принимает три различных значения. Эти наборы порождают куб размера 2, т. е. куб, имеющий проекцию на каждую ось не более чем из двух точек, а размерность куба не более n. Данный куб определяется как

$$\{\alpha, \beta\} \times \{\alpha_2, \gamma_2\} \times \ldots \times \{\alpha_n, \gamma_n\}.$$

Рассмотрим в этом кубе его гиперплоскости $x_1 = \alpha$ и $x_1 = \beta$ (см. рис. 2). В гиперплоскости $x_1 = \alpha$ соединим точку $(\alpha_2, \ldots, \alpha_n)$ с точкой $(\gamma_2, \ldots, \gamma_n)$ цепочкой ребер, принадлежащих этой гиперплоскости (каждое ребро — пара соседних наборов, т. е. наборов, принадлежащих гиперплоскости и отличающихся значением ровно в одной координате). Данная цепочка ребер определяет цепочку ребер в гиперплоскости $x_1 = \beta$, являющуюся ее проекцией. Пара соответствующих ребер этих цепочек образует квадрат. Следовательно, мы имеем цепочку квадратов. На ребре R_1 первого квадрата функция принимает значения $f(\alpha, \alpha_2, \ldots, \alpha_n)$ и $f(\beta, \alpha_2, \ldots, \alpha_n)$, а на ребре R_t последнего квадрата функция f не принимает хотя бы одно из этих значений. В таком случае в цепочке найдется квадрат (пусть его номер i) такой, что на ребре R_i функция принимает значения $f(\alpha, \alpha_2, \ldots, \alpha_n)$ и $f(\beta, \alpha_2, \ldots, \alpha_n)$, а на ребре R_{i+1} не принимает по крайней мере одно из этих значений. Квадрат с номером i является искомым. Лемма доказана.

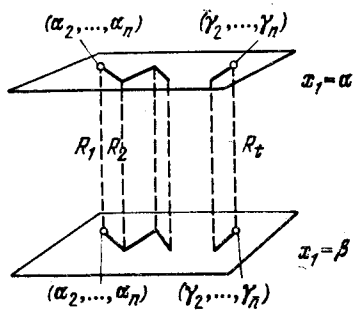


Рис. 2

Доказанные леммы допускают простую геометрическую интерпретацию. Пусть $f(x_1, \ldots, x_n)$ — существенная функция, принимающая l значений, где $l \geqslant 3$. Тогда:

1. Существует куб размера 2 такой, что на нем функция f принимает по крайней мере три значения (лемма 3).

2. Существует куб размера $l - 1$ такой, что на нем функция f принимает все l значений (лемма 4).

3. Существует квадрат, на котором функция f принимает либо более двух значений, либо ровно два, из которых одно в одной точке (лемма 5).

Замечание 1. Леммы 3 и 4 не имеют смысла для $k = 2$, лемма 5 имеет смысл, но неверна, так как функция

$$f(x_1, x_2) = x_1 + x_2$$

в своей области определения — а она есть квадрат,— принимает два значения, и оба с кратностью два.

Замечание 2. Леммы 4 и 5 не допускают усилений. В самом деле, пусть $3 \leqslant l \leqslant k-1$, $n \geqslant 3$ и

$$f_l(x_1, \ldots, x_n) = \begin{cases} i & \text{при } x_1 = \ldots = x_n = i,\ i \leqslant l-1, \\ 0 & \text{в остальных точках.} \end{cases}$$

Пусть $G_1, \ldots, G_n$ — произвольные множества из E_k и

$$1 \leqslant |G_1|, \ldots, |G_n| \leqslant l-2.$$

Тогда на $G_1 \times \ldots \times G_n$ функция f_l не может принимать все l значений. Далее, на любом квадрате f_l принимает не более двух значений.

При рассмотрении функций $f(x_1, \ldots, x_n)$ из P_k, обладающих определенными свойствами на некотором множестве $\mathscr{E} = G_1 \times \ldots \times G_n$, часто приходится переходить к функциям $f'(x_1, \ldots, x_n)$, обладающим аналогичными свойствами, но, может быть, на другом множестве $\mathscr{E}' = G_1' \times \ldots \times G_n'$. Переход от функции f к функции f' мы будем называть *нормировкой*. Следовательно, нормировка связана с преобразованием переменных и преобразованием значений функции вида

$$f'(x_1, \ldots, x_n) = \psi(f(\psi_1(x_1), \ldots, \psi_n(x_n))).$$

В этих преобразованиях мы исходим из того, что $|G_1'| = |G_1| = l_1, \ldots, |G_n'| = |G_n| = l_n$. Обозначим через $\eta_0, \ldots, \eta_{l-1}$ значения, которые принимает f на множестве $\mathscr{E}$, и через $\eta_0', \ldots, \eta_{l-1}'$ — набор из попарно различных чисел множества E_k. Таким образом, нормировка определяется указанием взаимно однозначных соответствий между множествами:

$$\{\eta_0, \ldots, \eta_{l-1}\} \leftrightarrow \{\eta_0', \ldots, \eta_{l-1}'\},$$

$$G_1 \leftrightarrow G_1',$$

$$\ldots\ldots$$

$$G_n \leftrightarrow G_n'.$$

Эти взаимно однозначные соответствия могут быть подчинены дополнительному требованию, например, чтобы фиксированная точка $(\alpha_1, \ldots, \alpha_n)$ из $\mathscr{E}$ соответствовала фиксированной точке $(\alpha_1', \ldots, \alpha_n')$ из $\mathscr{E}'$ и чтобы η_i соответствовало η_j'. Ясно, что эти соответствия могут быть осуществлены при помощи функций $\psi(x)$, $\psi_1(x)$, ...

..., $\psi_n(x)$ из P_k, которые всегда можно выбрать из множества подстановок; если $l, l_1, \ldots, l_n \leqslant k-1$, то их можно выбрать из множества функций одной переменной, принимающих не более $k-1$ значений. Очевидно, что при нормировке кратности значений функции f' на $\mathscr{E}'$ такие же, как кратности соответствующих значений функции f на $\mathscr{E}$. В дальнейшем нормировка используется в следующих видах:

1) $\{\eta_0, \ldots, \eta_{l-1}\} = \{\eta'_0, \ldots, \eta'_{l-1}\}$, $\psi(x) = x$,
$G'_i = \{0, \ldots, l_i - 1\}$ $(i = 1, \ldots, n)$.

2) $G_i = G'_i$, $\psi_i(x) = x$ $(i = 1, \ldots, n)$,
$\{\eta'_0, \ldots, \eta'_{l-1}\} = \{0, \ldots, l-1\}$.

3) $\{\eta'_0, \ldots, \eta'_{l-1}\} = \{0, \ldots, l-1\}$,
$G'_i = \{0, \ldots, l_i - 1\}$ $(i = 1, \ldots, n)$.

Случаи 1 (преобразование переменных) и 2 (преобразование значений) — случаи неполной нормировки. В качестве фиксированных точек η'_j и $(\alpha'_1, \ldots, \alpha'_n)$ обычно берутся 0 и $(0, \ldots, 0)$.

Докажем теперь теорему, являющуюся обобщением известной теоремы Слупецкого ([50]).

Теорема 7. *Пусть система $\mathfrak{P}$ функций из P_k, где $k \geqslant 3$, содержит все функции одной переменной, принимающие не более $k-1$ значений. Тогда для полноты системы $\mathfrak{P}$ необходимо и достаточно, чтобы $\mathfrak{P}$ содержала существенную функцию $f(x_1, \ldots, x_n)$, принимающую все k значений.*

Доказательство. Необходимость. Пусть $\mathfrak{P}$ — полная система. Предположим, что $\mathfrak{P}$ не содержит существенной функции, принимающей все k значений. Очевидно, что тогда из $\mathfrak{P}$ нельзя получить существенной функции, принимающей все k значений. Мы пришли к противоречию. Значит $\mathfrak{P}$ содержит существенную функцию, принимающую все k значений.

Достаточность. Пусть $\mathfrak{P}$ удовлетворяет условию теоремы и содержит существенную функцию $f(x_1, \ldots, x_n)$, принимающую все k значений. Покажем, пользуясь индукцией, что $\mathfrak{P}$ полна.

1) Базис индукции. Покажем, что из $\mathfrak{P}$ можно получить все функции из P_k, принимающие два значения.

На основании леммы 5 найдется квадрат

$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n),$$

где $\alpha_i \neq \beta_i$ и $\alpha_j \neq \beta_j$, на котором функция f принимает не менее двух значений, причем одно из них, η,— в одной точке. Возьмем функцию $\varphi(x)$ такую, что

$$\varphi(x) = \begin{cases} 0 & \text{при } x = \eta, \\ 1 & \text{при } x \neq \eta. \end{cases}$$

Очевидно, что $\varphi(x) \in \mathfrak{P}$. Пусть

$g(x_1, x_2) =$

$$= \varphi(f(\alpha_1, \ldots, \alpha_{i-1}, x_1, \alpha_{i+1}, \ldots, \alpha_{j-1}, x_2, \alpha_{j+1}, \ldots, \alpha_n)).$$

Функция $g(x_1, x_2)$ на квадрате $\{(\alpha_i, \alpha_j), (\beta_i, \alpha_j), (\beta_i, \beta_j), (\alpha_i, \beta_j)\}$ принимает два значения, 0 и 1, причем значение 0 в одной точке; обозначим ее (α_1^0, α_2^0). Осуществляя неполную нормировку так, чтобы (0, 0) отображалось в (α_1^0, α_2^0), а квадрат $\{(0, 0), (1, 0), (1,1), (1, 0)\}$ — в указанный выше квадрат, получим функцию $g'(x_1, x_2)$, где

$$g'(x_1, x_2) = g(\psi_1(x_1), \psi_2(x_2))$$

и $\psi_1, \psi_2 \in \mathfrak{P}$. Функция $g'(x_1, x_2)$, очевидно, есть максимум на множестве $\{0, 1\} \times \{0, 1\}$. Обозначим ее через $x_1 \vee_{01} x_2$. Так как система $\mathfrak{P}$ содержит функции $j_i(x)$, где

$$j_i(x) = \begin{cases} 1 & \text{при } x = i, \\ 0 & \text{при } x \neq i, \end{cases}$$

то

$$x_1 \&_{01} x_2 = j_0(j_0(x_1) \vee_{01} j_0(x_2))$$

есть минимум на множестве $\{0, 1\} \times \{0, 1\}$. Пусть $h(x_1, \ldots, x_m)$, $h \neq \text{const}$,— произвольная функция, принимающая два значения, 0 и 1; тогда

$h(x_{11} \ldots, x_m) =$

$$= \bigvee_{(\sigma_1, \ldots, \sigma_m)} j_{\sigma_1}(x_1) \& \ldots \& j_{\sigma_m}(x_m) \& h(\sigma_1, \ldots, \sigma_m) =$$

$$= \bigvee_{(\sigma_1, \ldots, \sigma_m)}^{01} j_{\sigma_1}(x_1) \&_{01} \ldots \&_{01} j_{\sigma_m}(x_m) \&_{01} h(\sigma_{11} \ldots, \sigma_m).$$

Таким образом, функция $h(x_1, \ldots, x_m)$ может быть получена из системы $\mathfrak{P}$. Так как $\mathfrak{P}$ содержит все функции одной переменной, принимающие любые два значения, то мы можем также получить из $\mathfrak{P}$ все функции, принимающие любые два значения.

2) Пусть из $\mathfrak{P}$ построены все функции, принимающие не более $l-1$ значений, $l-1<k$. Покажем, что тогда можно построить все функции из P_k, принимающие l значений.

Возьмем функцию $f(x_1, \ldots, x_n)$. На основании леммы 4 найдутся n подмножеств $G_1, \ldots, G_n$ таких, что $|G_i| \leqslant l-1$ $(i=1, \ldots, n)$, и на $\mathscr{E}=G_1\times\ldots\times G_n$ функция f принимает l значений $\eta_0, \eta_1, \ldots, \eta_{l-1}$. Пусть эти значения принимаются соответственно на наборах из $\mathscr{E}$:

$$\tilde{\alpha}^{(0)}=(\alpha_1^{(0)}, \alpha_2^{(0)}, \ldots, \alpha_n^{(0)}),$$
$$\tilde{\alpha}^{(1)}=(\alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_n^{(1)}),$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$\tilde{\alpha}^{(l-1)}=(\alpha_1^{(l-1)}, \alpha_2^{(l-1)}, \ldots, \alpha_n^{(l-1)}),$$

т. е.

$$f(\tilde{\alpha}^{(0)})=\eta_0, \quad f(\tilde{\alpha}^{(1)})=\eta_1, \ldots, f(\tilde{\alpha}^{(l-1)})=\eta_{l-1}.$$

Покажем, что из системы $\mathfrak{P}$ можно построить произвольную функцию $h(x_1, \ldots, x_m)$, принимающую значения $\eta_0, \eta_1, \ldots, \eta_{l-1}$.

В самом деле, функцию $h(x_1, \ldots, x_m)$ можно задать при помощи табл. 2, в которой $\tilde{\sigma}=(\sigma_1, \ldots, \sigma_m)$. Определим функции $\psi_j(x_1, \ldots, x_m)$ $(j=1, \ldots, n)$ так, как указано в табл. 3.

Таблица 2

$x_1 \ldots x_m$	$h(x_1, \ldots, x_m)$
$\sigma_1 \ldots \sigma_m$	$\eta_{i(\tilde{\sigma})}$

Таблица 3

$x_1 \ldots x_m$	$\psi_j(x_1, \ldots, x_m)$
$\sigma_1 \ldots \sigma_m$	$\alpha_j^{(i(\tilde{\sigma}))}$

Тогда

$$h(x_1, \ldots, x_m)=f(\psi_1(x_1, \ldots, x_m), \ldots, \psi_n(x_1, \ldots, x_m)),$$

ибо

$$f(\psi_1(\sigma_1, \ldots, \sigma_m), \ldots, \psi_n(\sigma_1, \ldots, \sigma_m)) =$$
$$= f\left(\alpha_1^{(i(\tilde{\sigma}))}, \ldots, \alpha_n^{(i(\tilde{\sigma}))}\right) = \eta_{i(\tilde{\sigma})},$$
$$h(\sigma_1, \ldots, \sigma_m) = \eta_{i(\tilde{\sigma})}.$$

Имея все функции с заданными l значениями $\eta_0, \ldots, \eta_{l-1}$, можно в случае $l < k$ получить при помощи функций одной переменной, принимающих менее k значений, остальные функции с l значениями. Таким образом, применяя этот прием, мы дойдем до $l = k$, и тогда построим все функции из P_k. Этим достаточность доказана.

Следствие (критерий Слупецкого). *Пусть система $\mathfrak{P}$ функций из P_k, где $k \geqslant 3$, содержит все функции одной переменной. Тогда для полноты системы $\mathfrak{P}$ необходимо и достаточно, чтобы $\mathfrak{P}$ содержала существенную функцию $f(x_1, \ldots, x_n)$, принимающую все k значений.*

Замечание. Доказанная теорема верна при $k \geqslant 3$ и не может быть распространена на случай $k = 2$. В самом деле, система

$$\mathfrak{P} = \{0, 1, x, \bar{x}, x_1 + x_2\}$$

не является полной, так как $\mathfrak{P} \subseteq L$.

Непосредственное использование теоремы и тем более ее следствия не всегда удобно, так как для этого предварительно нужно установить наличие в $\mathfrak{P}$ всех функций одной переменной, принимающих не более $k - 1$ значений, т. е. $k^k - k!$ функций. С ростом k громоздкость указанных построений сильно возрастает. Поэтому целесообразно в формулировках теорем заменить требование, чтобы система $\mathfrak{P}$ содержала определенное множество функций одной переменной, требованием, чтобы система $\mathfrak{P}$ порождала это множество функций одной переменной. Последнее может быть установлено гораздо более экономичным образом. Например, если известно, что из $\mathfrak{P}$ могут быть получены какие-то конкретные системы функций одной переменной, порождающие все функции одной переменной.

В качестве примеров таких систем приведем две системы.

Теорема 8 (С. Пикар [45]). *Все функции одной переменной из P_k могут быть порождены тремя функ-*

циями:

$$f(x) = x - 1 \pmod{k},$$

$$g(x) = \begin{cases} x, & 0 \leqslant x \leqslant k-3, \\ k-1, & x = k-2, \\ k-2, & x = k-1, \end{cases} \qquad h(x) = \begin{cases} 1, & x = 0, \\ x, & x \neq 0. \end{cases}$$

Теорема 9. *Все функции одной переменной из P_k могут быть порождены k функциями*

$$f_i(x) = \begin{cases} i & \text{при } x = 0, \\ 0 & \text{при } x = i, \\ x & \text{в остальных случаях} \end{cases} \quad (i = 1, \ldots, k-1)$$

и функцией $h(x)$.

Доказательства этих теорем несложны, поэтому они опускаются.

В заключение рассмотрим еще одно приложение доказанного критерия полноты. Мы дадим характеристическое свойство функции из P_k, образующей полную систему (функция Шеффера). Это свойство является незначительным усилением теоремы Мартина [44].

Теорема 10. *Функция $f(x_1, \ldots, x_n)$ из P_k, где $k \geqslant 3$, является функцией Шеффера тогда и только тогда, когда $f(x_1, \ldots, x_n)$ порождает все функции одной переменной, принимающие не более $k-1$ значений.*

Доказательство. Необходимость очевидна.

Достаточность. Очевидно, $f(x_1, \ldots, x_n)$ должна принимать все k значений, так как из нее получаются, например, все константы. Если f — несущественная функция, то f есть подстановка и из нее можно получить только подстановки. В этом случае мы не получим даже констант. Значит, это невозможно, т. е. f является существенной функцией. Применяя критерий полноты, приходим к тому, что система $\mathfrak{P} = \{f(x_1, \ldots, x_n)\}$ является полной. Теорема доказана.

§ 5. Особенности k-значных логик

Предыдущий материал показывает, что во многом конечнозначные логики похожи на двузначную логику. В них сохраняются многие результаты, имеющие место в двузначной логике. Правда, рост значности все-таки

приводит к известным усложнениям формулировок и доказательств.

Однако теперь уже накоплено достаточно много фактов, указывающих на своеобразие конечнозначных логик, в том числе и фактов, выявляющих существенное отличие P_k при $k \geqslant 3$ от P_2. Эти обстоятельства тем более заслуживают внимания, если иметь в виду работу Поста [46], в которой была выдвинута идея сведения конечнозначных логик к двузначной логике. Он предложил вместо функции $f(x_1, \ldots, x_n)$ из P_k рассматривать систему функций

$$\varphi_1(x_{11}, \ldots, x_{1l}, \ldots, x_{n1}, \ldots, x_{nl}), \ldots$$
$$\ldots, \varphi_l(x_{11}, \ldots, x_{1l}, \ldots, x_{n1}, \ldots, x_{nl}),$$

где $l =]\log_2 k[$ *), причем если значение α_i имеет двоичный код $\alpha_{i1} \ldots \alpha_{il}$ $(i = 1, \ldots, n)$, то значение $f(\alpha_1, \ldots, \alpha_n)$ имеет двоичный код

$$\varphi_1(\alpha_{11}, \ldots, \alpha_{1l}, \ldots, \alpha_{n1}, \ldots, \alpha_{nl}) \ldots$$
$$\ldots \varphi_l(\alpha_{11}, \ldots, \alpha_{1l}, \ldots, \alpha_{n1}, \ldots, \alpha_{nl}).$$

При этом операции суперпозиции в P_k будет соответствовать весьма специальная операция над системой функций $\{\varphi_1, \ldots, \varphi_l\}$ из P_2. Возможность кодирования функций из P_k системами функций из P_2 может быть полезной при решении логических задач на ЭВМ, но мало что дает для исследований в конечнозначных логиках. Факты, свидетельствующие о своеобразии P_k при $k \geqslant 3$, были известны, начиная с работ Слупецкого [50], а также работ Кузнецова [15] и Яблонского [35, 36]. Однако более поздние результаты показали, что различие между P_k и P_2 значительно существеннее, чем это казалось раньше.

В настоящем параграфе мы собираемся коснуться лишь некоторых сторон этой проблемы. Нас будут интересовать следующие три вопроса.

1. Вопрос о существовании базисов для замкнутых классов в P_k.

2. Вопрос о мощности системы всех замкнутых классов в P_k.

3. Вопрос о возможности представления функций из P_k посредством полиномов.

---

*) $]a[$ обозначает наименьшее целое число, не меньшее a.

Как это вытекает из теорем Поста и Жегалкина, в алгебре логики мы имеем следующие ответы на поставленные вопросы.

1. Каждый замкнутый класс в P_2 имеет конечный базис.

2. Мощность множества всех замкнутых классов в P_2 равна $\aleph_0$.

3. Всякая функция в P_2 может быть записана в виде полинома по mod 2.

Для P_k $(k \geqslant 3)$ соответствующие ответы составляют содержание последующих теорем. Первая из них есть теорема Янова.

Теорема 11 (Янов [39]). *Для всякого k $(k \geqslant 3)$ существует в P_k замкнутый класс, не имеющий базиса.*

Доказательство. Рассмотрим последовательность функций

$$f_0 = 0,$$

$$f_i(x_1, \ldots, x_i) = \begin{cases} 1 \text{ при } x_1 = \ldots = x_i = 2, \\ 0 \text{ в остальных случаях} \end{cases} \quad (i = 1, 2, \ldots).$$

Обозначим через $\mathfrak{M}_k$ множество всех функций, получающихся из $\{f_0, f_1, \ldots\}$ путем переименования (без отождествления) переменных. Легко видеть, что $\mathfrak{M}_k$ — замкнутый класс. Допустим, что $\mathfrak{M}_k$ имеет базис. Тогда в базисе найдется функция $\bar{f}$, получающаяся из функции f_{n_0} путем переименования переменных, для которой число n_0 минимально. Далее возможны два случая.

1. Базис содержит еще хотя бы одну функцию $\bar{f}'$. Этой функции соответствует функция f_{n_1} и $n_1 > n_0$. Так как f_{n_0} может быть получена из f_{n_1} путем отождествления переменных, то $\bar{f}$ выражается через $\bar{f}'$, что противоречит определению базиса.

2. Базис состоит из единственной функции $\bar{f}$. В этом случае никакая функция f_n при $n > n_0$ не может быть получена из $\bar{f}$, так как $f_{n_0}(\ldots, f_{n_0}, \ldots) \equiv 0$. Мы опять приходим к противоречию.

Итак, остается допустить, что $\mathfrak{M}_k$ не имеет базиса. Теорема доказана.

Теорема 12 (Мучник [39]). *Для всякого k $(k \geqslant 3)$ существует в P_k замкнутый класс со счетным базисом.*

Доказательство. Рассмотрим последовательность функций ($i = 2, 3, \ldots$)

$$f_i(x_1, \ldots, x_i) =$$
$$= \begin{cases} 1 \text{ при } x_1 = \ldots = x_{j-1} = x_{j+1} = \ldots = x_i = 2,\ x_j = 1 \\ \qquad\qquad (j = 1, \ldots, i), \\ 0 \text{ в остальных случаях.} \end{cases}$$

Обозначим через $\mathfrak{M}_k$ замкнутый класс, порожденный системой $\{f_2, f_3, \ldots\}$. Покажем, что эта система является базисом в $\mathfrak{M}_k$. Для доказательства достаточно установить, что никакая из функций f_m не может быть выражена в виде формулы через остальные функции системы, т. е. что невозможно представление

$$f_m = \mathfrak{A}[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots].$$

Если записать формулу $\mathfrak{A}$ несколько подробнее, то получим

$$\mathfrak{A}[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots] =$$
$$= f_r(\mathfrak{B}_1[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots], \ldots$$
$$\ldots, \mathfrak{B}_r[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots])$$

или

$$f_m(x_1, \ldots, x_m) =$$
$$= f_r(\mathfrak{B}_1[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots], \ldots$$
$$\ldots, \mathfrak{B}_r[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots]).$$

Априори возможны три случая.

1. Среди формул $\mathfrak{B}_1, \ldots, \mathfrak{B}_r$ (здесь $r \geqslant 2$) по крайней мере две формулы отличны от символов переменных. Тогда при любых значениях переменных $x_1, \ldots, x_m$ на соответствующих местах функции f_r возможны лишь значения 0 и 1 и поэтому правая часть будет тождественно равна нулю. Последнее противоречит возможности такого представления, так как $f_m \not\equiv 0$.

2. Среди формул $\mathfrak{B}_1, \ldots, \mathfrak{B}_r$ найдется только одна формула $\mathfrak{B}_s$, которая отлична от символа переменной. По условию остальные формулы сводятся к переменным, и поскольку $r \geqslant 2$, то найдется по крайней мере одна формула $\mathfrak{B}_p \equiv x_q$. Рассмотрим набор $x_1 = \ldots = x_{q-1} = x_{q+1} = \ldots = x_m = 2$ и $x_q = 1$. На этом наборе формула $\mathfrak{B}_s$ принимает значение либо 0, либо 1. Следовательно, при данном выборе значений переменных у функции f_r на двух ме-

стах будут стоять значения, отличные от 2. Поэтому правая часть примет значение 0. В то же время левая часть на данном наборе равна 1. Мы пришли к противоречию.

3. Все формулы $\mathfrak{B}_1, \ldots, \mathfrak{B}_r$ — символы переменных. В этом случае $r > m$ и, следовательно, в формуле встретятся по крайней мере два вхождения некоторой переменной x_p. Взяв набор $x_1 = \ldots = x_{p-1} = x_{p+1} = \ldots = x_m = 2$ и $x_p = 1$, мы обратим левую часть в 1, а правую в 0. Следовательно, этот случай также невозможен. Теорема доказана.

Непосредственно к доказанному примыкает следующая теорема.

Теорема 13. *Для всякого k ($k \geqslant 3$) P_k содержит континуум различных замкнутых классов.*

Доказательство. Число замкнутых классов в P_k можно оценить сверху числом всех подмножеств функций из P_k. Так как P_k содержит счетное число функций, то число подмножеств P_k равно континууму.

Для завершения доказательства нужно оценить снизу число замкнутых классов в P_k. С этой целью рассмотрим замкнутый класс $\mathfrak{M}_k$, построенный при доказательстве предыдущей теоремы. Этот класс имеет базис

$$\{f_2, f_3, \ldots\}.$$

Образуем для каждой последовательности $\{\rho_1, \rho_2, \ldots\}$, где $2 \leqslant \rho_1 < \rho_2 < \ldots$, класс $\mathfrak{M}_k(\rho_1, \rho_2, \ldots)$ как класс, порожденный системой функций $\{f_{\rho_1}, f_{\rho_2}, \ldots\}$. Легко видеть, что

$$\mathfrak{M}_k(\rho'_1, \rho'_2, \ldots) \neq \mathfrak{M}_k(\rho''_1, \rho''_2, \ldots),$$

если

$$\{\rho'_1, \rho'_2, \ldots\} \neq \{\rho''_1, \rho''_2, \ldots\}.$$

Следовательно, семейство $\{\mathfrak{M}_k(\rho_1, \rho_2, \ldots)\}$ является континуальным семейством. Теорема доказана.

Теорема 14. *Система полиномов по* $\bmod k$ *полна в P_k тогда и только тогда, когда $k = p$, где p — простое число.*

Доказательство. Легко видеть, что для любой функции $f(x_1, \ldots, x_n)$ из P_k имеет место представление

$$f(x_1, \ldots, x_n) = \sum_{(\sigma_1, \ldots, \sigma_n)} j_{\sigma_1}(x_1) \ldots j_{\sigma_n}(x_n) f(\sigma_1, \ldots, \sigma_n) \pmod k.$$

Поэтому вопрос о представимости функции f полиномами по $\bmod k$ сводится к вопросу о представимости в виде полиномов функций

$$j_0(x),\ \ldots,\ j_{k-1}(x).$$

В силу того, что

$$j_\sigma(x) = j_0(x-\sigma),$$

мы можем утверждать, что система полиномов по $\bmod k$ полна тогда и только тогда, когда представима в виде полинома функция $j_0(x)$.

1. Пусть $k = p$. В этом случае, опираясь на малую теорему Ферма $a^{p-1} \equiv 1 \pmod p$ $(1 \leqslant a \leqslant p-1)$, получаем

$$j_0(x) = 1 - x^{p-1} \pmod p,$$

т. е. система полиномов полна в P_k*).

Можно указать другой способ решения этой же задачи. Будем искать представление функции $g(x)$, зависящей от одной переменной, в виде полинома, пользуясь методом неопределенных коэффициентов

$$g(x) = a_0 + a_1 x + \ldots + a_{p-1}x^{p-1}.$$

Мы получаем систему уравнений

$$\begin{aligned}
&a_0 + a_1 \cdot 0^1 + a_2 \cdot 0^2 + \ldots + a_{p-1} \cdot 0^{p-1} = g(0),\\
&a_0 + a_1 \cdot 1^1 + a_2 \cdot 1^2 + \ldots + a_{p-1} \cdot 1^{p-1} = g(1),\\
&a_0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + \ldots + a_{p-1} \cdot 2^{p-1} = g(2),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_0 + a_1(p-1)^1 + a_2(p-1)^2 + \ldots + a_{p-1}(p-1)^{p-1} =\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = g(p-1).
\end{aligned}$$

Определитель этой системы

$$\Delta = \begin{vmatrix}
1 & 0 & 0 & \ldots & 0\\
1 & 1 & 1^2 & \ldots & 1^{p-1}\\
1 & 2 & 2^2 & \ldots & 2^{p-1}\\
\ldots & \ldots & \ldots & \ldots & \ldots\\
1 & p-1 & (p-1)^2 & \ldots & (p-1)^{p-1}
\end{vmatrix}$$

есть определитель Вандермонда. Как известно,

$$\Delta = \prod_{0 \leqslant i < j \leqslant p-1} (j-i).$$

---

*) Малая теорема Ферма обосновывается так. Пусть $1 \leqslant a \leqslant p-1$, тогда числа $r_1 = a \cdot 1,\ \ldots,\ r_{p-1} = a(p-1)$ не сравнимы по $\bmod p$. Поэтому $r_1 \ldots r_{p-1} \equiv (p-1)! \pmod p$ и $(p-1)! \equiv a^{p-1}(p-1)! \pmod p$ или $1 \equiv a^{p-1} \pmod p$.

Так как p — простое число, то $\Delta \not\equiv 0 (\mathrm{mod}\, p)$. Пользуясь правилом Крамера и учитывая, что $\Delta \not\equiv 0 (\mathrm{mod}\, p)$, мы сможем решить в целых числах сравнения

$$a_i \Delta \equiv \Delta_i (\mathrm{mod}\, p) \quad (i = 0, \ldots, p-1),$$

где Δ_i — соответствующий минор. Итак, мы приходим к единственному решению исходной системы и, следовательно, к полиному, изображающему $g(x)$.

2. Пусть $k \neq p$. Тогда $k = k_1 k_2$, где $k > k_2 > 1$. Допустим, что

$$j_0(x) = b_0 + b_1 x + \ldots + b_s x^s \ (\mathrm{mod}\, k).$$

При $x = 0$ получаем $b_0 = 1$. При $x = k_1$ получаем

$$0 = 1 + b_1 k_1 + \ldots + b_s k_1^s \quad (\mathrm{mod}\, k)$$

или

$$k - 1 = b_1 k_1 + \ldots + b_s k_1^s \quad (\mathrm{mod}\, k),$$

т. е. $k - 1$ делится на k_1. Таким образом, k и $k - 1$ делятся на k_1, что возможно только при $k_1 = 1$. Мы пришли к противоречию. Следовательно, при $k \neq p$ функция $j_0(x)$ не представима полиномом по mod k.

Доказанная теорема может быть легко обобщена на случай, когда на E_k возможно определить две операции: $\oplus$ и $\times$ — сложение и умножение, относительно которых E_k образует поле. Как показывается в алгебре, конечное поле или поле Галуа, существует тогда и только тогда, когда $k = p^m$. В этом случае оно определяется с точностью до изоморфизма однозначным образом. При этом относительно сложения оно образует абелеву группу характеристики p, т. е. для любого элемента α выполняется соотношение

$$\underbrace{\alpha \oplus \ldots \oplus \alpha}_{p} = 0,$$

где 0 — нуль группы. Эту группу можно определить, рассматривая числа α, как числа в p-ичной системе счисления, т. е. в виде наборов $(\alpha_1, \ldots, \alpha_m)$, и операцию $\alpha \oplus \beta = (\alpha_1 + \beta_1, \ldots, \alpha_m + \beta_m)$ ($+$ обозначает сложение по mod p). Все элементы поля Галуа, кроме 0, образуют относительно второй операции циклическую группу.

Пример 5. Пусть $k = 2^2$. Тогда операция $\oplus$ имеет вид как в табл. 4. Для построения таблицы для операции $\times$ заметим, что числа 1, 2, 3 могут быть выражены

как степени некоторого элемента α (следует из цикличности мультипликативной группы). Это число α удовлетворяет уравнению

$$\alpha^3 = 1$$

или, так как $\alpha \neq 1$,

$$\alpha^2 \oplus \alpha \oplus 1 = 0.$$

Взяв за α элемент 2 получим $\alpha^2 = \ominus(\alpha \oplus 1) = 3$. Мы получаем таблицу для $\times$ (табл. 5), поскольку

$$2 \cdot 2 = \alpha \cdot \alpha = 3,$$

$$2 \cdot 3 = \alpha \cdot \alpha^2 = \alpha^3 = 1,$$

$$3 \cdot 3 = \alpha^2 \cdot \alpha^2 = \alpha^4 = \alpha = 2.$$

Мы можем, повторяя первую часть доказательства предыдущей теоремы, показать, что каждая функция $f(x_1, \ldots, x_n)$ из P_k при $k = p^m$ представима полиномом над соответствующим полем Галуа.

Таблица 4

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Таблица 5

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

В частности, в P_4 возможно представление функций полиномами, но не по mod 4, а полиномами над полем Галуа.

Итак, для P_k ($k \geqslant 3$) мы получаем следующие ответы на поставленные в начале параграфа вопросы.

1. Существуют в P_k замкнутые классы, не имеющие конечного базиса.

2. Мощность множества всех замкнутых классов в P_k равна $\mathfrak{c}$.

3. Всякая функция в P_k может быть записана в виде полинома по mod k (соответственно над полем Галуа) в том и только том случае, когда $k = p$ (соответственно, когда $k = p^m$).

Сопоставляя эти ответы с ответами для двузначного случая, мы видим, насколько существенно различие указанных логик. Кроме того, мы видим, что некоторые вопросы решаются по-разному в зависимости от значения числа k.

Глава 3

ОГРАНИЧЕННО-ДЕТЕРМИНИРОВАННЫЕ (АВТОМАТНЫЕ) ФУНКЦИИ С ОПЕРАЦИЯМИ

Мы познакомились с двумя функциональными системами с операциями:

(P_2, C) — алгебра логики — система функций алгебры логики с операцией суперпозиции;

(P_k, C) — k-значная логика — система функций k-значной логики с операцией суперпозиции.

Путем вариации функционального объекта и операции можно получать другие системы. Так, усложняя функциональные объекты, естественным образом получаем:

$(P_{\aleph_0}, C)$ — счетнозначную логику, т. е. систему, содержащую константы $0, 1, \ldots, k, \ldots$ и функции $f(x_1, \ldots, x_n)$, переменные которых определены на расширенном натуральном ряде $E_{\aleph_0} = \{0, 1, 2, \ldots\}$, а сами функции принимают значения на $E_{\aleph_0}$, с операцией суперпозиции;

(P_c, C) — континуумзначную логику, т. е. систему, содержащую константы из [0, 1] и функции, переменные которых определены на сегменте [0, 1] и сами принимают значения на [0, 1], с операцией суперпозиции.

Мы не будем подробно рассматривать эти две системы, а познакомимся с другими, более важными. В этой главе речь будет идти о функциональной системе, связанной с автоматами.

§ 1. Детерминированные функции

Функциональный объект, который мы будем рассматривать, является разновидностью континуумзначной логики. Вместо действительных чисел из сегмента [0, 1] мы возьмем множество $E_{c,k}$ всех k-значных последователь-

ностей α, где

$$\alpha = \{\alpha(1), \alpha(2), \ldots, \alpha(m), \ldots\},$$
$$\alpha(m) \in E_k \text{ для всех } m \ (m = 1, 2, \ldots).$$

Обозначим через $P_{\text{с},k}$ множество всех функций

$$f(x_1, \ldots, x_n),$$

определенных на наборах $(\alpha_1, \ldots, \alpha_n)$, где $\alpha_i \in E_{\text{с},k}$ $(i = 1, 2, \ldots, n)$, и принимающих значения из $E_{\text{с},k}$. Таким образом, функции из $P_{\text{с},k}$ преобразуют наборы k-значных последовательностей в k-значные последовательности. В $P_{\text{с},k}$ включим также все последовательности из $E_{\text{с},k}$, рассматривая их как функции, зависящие от пустого множества переменных $(n = 0)$, т. е. как константы.

Пример 1. Пусть $k = 2$ и

$$f(\alpha) = \begin{cases} (0, 0, \ldots), & \text{если } \alpha = (0, 0, \ldots), \\ (1, 1, \ldots), & \text{если } \alpha \neq (0, 0, \ldots). \end{cases}$$

Очевидно, что

$$f(x) \in P_{\text{с},2}.$$

Заметим, что для функции $P_{\text{с},k}$ табличное задание неприемлемо, так как множество $E_{\text{с},k}$ (а следовательно, и множество «строк» таблицы) имеет континуальную мощность. Отсюда же следует, что мощность множества всех функций $P_{\text{с},k}$, зависящих от переменных $x_1, x_2, \ldots$, равна гиперконтинууму. Учитывая это обстоятельство, мы в дальнейшем будем рассматривать более узкий функциональный объект.

Для наборов и функций мы будем дальше употреблять векторную запись. Так, обозначая набор переменных $(x_1, x_2, \ldots, x_n)$ через X, вместо $f(x_1, \ldots, x_n)$ мы будем писать $f(X)$. При этом значение переменной X есть вектор (набор) $\alpha = (\alpha_1, \ldots, \alpha_n)$, компонентами которого являются последовательности значности k:

$$\alpha_i = \{\alpha_i(1), \alpha_i(2), \ldots, \alpha_i(m), \ldots\} \quad (i = 1, 2, \ldots, n).$$

Будем истолковывать α как последовательность векторов

$$\alpha = \{\alpha(1), \alpha(2), \ldots, \alpha(m), \ldots\},$$

где

$$\alpha(m)=(\alpha_1(m),\ \alpha_2(m),\ \ldots,\ \alpha_n(m)) \quad (m=1,\ 2,\ \ldots).$$

Таким образом, мы считаем, что выполнено тождество

$$\begin{aligned}(&\{\alpha_1(1),\ \alpha_1(2),\ \ldots,\ \alpha_1(m),\ \ldots\},\\ &\{\alpha_2(1),\ \alpha_2(2),\ \ldots,\ \alpha_2(m),\ \ldots\},\ \ldots\\ &\ldots,\ \{\alpha_n(1),\ \alpha_n(2),\ \ldots,\ \alpha_n(m),\ \ldots\})=\\ =&\{(\alpha_1(1),\ \alpha_2(1),\ \ldots,\ \alpha_n(1)),\ (\alpha_1(2),\ \alpha_2(2),\ \ldots\\ &\ldots,\ \alpha_n(2)),\ \ldots,\ (\alpha_1(m),\ \alpha_2(m),\ \ldots,\ \alpha_n(m)),\ \ldots\}.\end{aligned}$$

Полученную последовательность векторов можно рассматривать как последовательность наборов $(\alpha_1(m),\ \alpha_2(m),\ \ldots\ \ldots,\ \alpha_n(m))$ или чисел в k-ичной системе счисления. Каждое из этих чисел принадлежит множеству E_N, где $N=k^n$.

Итак, функцию $f(x_1,\ \ldots,\ x_n)$ из $P_{c,k}$ можно рассматривать как функцию $f(X)$ из множества $P_{c,N}$, но зависящую от одной переменной (и принимающую значения из $E_{c,k}\subset E_{c,N}$). Таким образом, изучение функции $f(x_1,\ \ldots\ \ldots,x_n)$ из $P_{c,k}$ можно свести к изучению функции $f(X)$ от одной переменной из $P_{c,N}$, где $N=k^n$. Данная редукция построена на формальных соображениях, связанных с толкованием набора последовательностей как последовательности наборов. Ниже мы увидим, что для некоторого класса функций из $P_{c,k}$ такое толкование приобретает определенный физический смысл.

Определение. Функция $f(X)$ из $P_{c,N}$ называется *детерминированной*, если каково бы ни было число m и каковы бы ни были последовательности α и β такие, что

$$\alpha(1)=\beta(1),\quad \alpha(2)=\beta(2),\ \ldots,\ \alpha(m)=\beta(m),$$

значения γ и δ функции f, где $\gamma=f(\alpha)$ и $\delta=f(\beta)$, представляют собой последовательности, у которых тоже совпадают первые m членов, т. е.

$$\gamma(1)=\delta(1),\quad \gamma(2)=\delta(2),\ \ldots,\ \gamma(m)=\delta(m).$$

Через $P_{д,k}$ обозначим множество всех детерминированных функций из $P_{c,k}$. $P_{д,k}$, очевидно, содержит все константы из $P_{c,k}$.

Пусть $f(\alpha)=\gamma$. Из определения следует, что у детерминированной функции значение $\gamma(m)$ m-го $(m=1,2,\ldots)$

члена последовательности γ полностью определяется значениями первых m членов

$$\alpha(1), \alpha(2), \ldots, \alpha(m)$$

последовательности α, т. е.

$$\gamma(m) = f_m(\alpha(1), \alpha(2), \ldots, \alpha(m)).$$

Поскольку

$$f_m(\alpha(1), \alpha(2), \ldots, \alpha(m)) = \\ = f_m(\alpha_1(1), \ldots, \alpha_n(1), \alpha_1(2), \ldots, \alpha_n(m)),$$

то ясно, что f_m — функция из P_k, зависящая от nm переменных.

Таким образом, детерминированная функция $f(X)$ определяется последовательностью функций k-значной логики

$$f \sim \{f_1, f_2, \ldots, f_m, \ldots\},$$

где

$$f_1 = f_1(X_1),$$
$$f_2 = f_2(X_1, X_2),$$
$$\ldots\ldots\ldots$$
$$f_m = f_m(X_1, X_2, \ldots, X_m),$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$X_1 = (x_1(1), x_2(1), \ldots, x_n(1)),$$
$$X_2 = (x_1(2), x_2(2), \ldots, x_n(2)),$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$X_m = (x_1(m), x_2(m), \ldots, x_n(m)),$$
$$\ldots\ldots\ldots\ldots\ldots$$

Детерминированная функция $f(x_1, \ldots, x_n)$ может быть проинтерпретирована следующим образом. Пусть мы имеем некоторый «дискретный преобразователь» (рис. 1), в котором существует n *входов* $x_1, x_2, \ldots, x_n$ и один *выход* f. На входы в *моменты времени* $t = 1, 2, \ldots, m, \ldots$ подаются (входные) последовательности

$$\alpha_1 = \{\alpha_1(1), \alpha_1(2), \ldots, \alpha_1(m), \ldots\},$$
$$\alpha_2 = \{\alpha_2(1), \alpha_2(2), \ldots, \alpha_2(m), \ldots\},$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$\alpha_n = \{\alpha_n(1), \alpha_n(2), \ldots, \alpha_n(m), \ldots\}.$$

И в эти же моменты t на выходе возникает (выходная) последовательность $\gamma = \{\gamma(1), \gamma(2), \ldots, \gamma(m), \ldots\}$, причем $\gamma = f(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Очевидно, что в дискретном преобразователе значение $\gamma(m)$ зависит только от значений входных последовательностей в моменты времени $t = 1, 2, \ldots, m$ и не зависит от значений в будущие моменты времени. Поэтому преобразование f есть детерминированная функция.

Рис. 1

Заметим, что константы из $P_{д,k}$ $(n = 0)$ интерпретируются дискретным преобразователем без входов («генератором»). Заметим также, что поступающие на входы преобразователя последовательности, т. е. $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, можно рассматривать как последовательность наборов

$$\{(\alpha_1(1), \alpha_2(1), \ldots, \alpha_n(1)), (\alpha_1(2), \alpha_2(2), \ldots, \alpha_n(2)), \ldots\}.$$

Здесь введенное нами тождество выполнено естественным образом.

Из того, что детерминированная функция $f(x_1, \ldots, x_n)$ полностью определена последовательностью функций k-значной логики, вытекает следующий факт.

Теорема 1. *Мощность множества всех детерминированных функций, зависящих от переменных* $x_1, x_2, \ldots, x_n$, *равна* $\mathfrak{c}$.

В заключение приведем ряд иллюстраций.

Пример 2. а) Функция $f_\Phi(x_1, \ldots, x_n)$, где $\Phi(x_1, \ldots, x_n) \in P_k$, определена следующим образом:

$$f_\Phi(x_1, \ldots, x_n) \sim \{\Phi(x_1(1), \ldots, x_n(1)), \Phi(x_1(2), \ldots, x_n(2)), \ldots, \Phi(x_1(m), \ldots, x_n(m)), \ldots\}.$$

Значение функции f_Φ определяется путем вычисления значений функции Φ по значениям соответствующих членов входных последовательностей. Отсюда $f_\Phi \in P_{д,k}$. В частности, если взять $\Phi(x_1, x_2) = x_1 \& x_2$ $(k = 2)$, то

$$f_{\&}(x_1, x_2) \sim \{x_1(1) \& x_2(1), x_1(2) \& x_2(2), \ldots, x_1(m) \& x_2(m), \ldots\}.$$

Здесь выходная последовательность — почленная конъюнкция входных последовательностей.

Обозначим через $\mathscr{P}_k$ множество всех функций f_Φ, где $\Phi \in P_k$.

б) Функция $z = x + y$, представляющая сложение двух k-значных последовательностей, определяется путем использования обычного алгоритма сложения двух чисел столбиком в k-ичном счислении, но только с бесконечным числом разрядов:

$$\begin{array}{rl} & \ldots x(3), x(2), x(1) \\ + & \ldots y(3), y(2), y(1) \\ \hline & \ldots z(3), z(2), z(1) \end{array}$$

Ясно, что $z(m)$ определяется по первым m членам слагаемых. Поэтому $x + y \in P_{д,k}$.

в) Функция x^2 определяется через алгоритм умножения чисел в k-ичной системе счисления, но с бесконечным числом разрядов

$$\begin{array}{rl} & \begin{array}{l} \times \ldots x(3), \quad x(2), \quad x(1) \\ \quad \ldots x(3), \quad x(2), \quad x(1) \end{array} \\ \hline & \begin{array}{l} \ldots x(3)\,x(1), x(2)\,x(1), x(1)\,x(1) \\ + \ldots x(2)\,x(2), x(1)\,x(2) \\ \quad \ldots x(1)\,x(3) \\ \quad \ldots\ldots\ldots\ldots\ldots\ldots \end{array} \\ \hline & \ldots z(3), \quad z(2), \quad z(1) \end{array}$$

Здесь $z(m)$ полностью определяется по первым m членам входной последовательности. Поэтому $x^2 \in P_{д,k}$.

Нетрудно привести примеры функций из $P_{с,k}$, которые не являются детерминированными. Так, функция $f(x)$ из примера 1 этого параграфа не является детерминированной.

Итак, на первом этапе мы получили подкласс $P_{д,k}$ класса $P_{с,k}$. Однако класс $P_{д,k}$ является также обширным — его мощность есть $\mathfrak{c}$.

§ 2. Задание детерминированных функций при помощи деревьев. Вес дерева

Для детерминированных функций можно предложить более наглядный способ задания, чем для произвольных функций из $P_{с,k}$. Он основан на аппарате деревьев *).

*) Строгое определение дерева приводится в части III.

Пусть k, n — целые числа и $N = k^n$. Рассмотрим бесконечную фигуру, изображенную на рис. 2. Она состоит из *вершин* и ориентированных *ребер*. Будем называть эту фигуру *деревом*. Вершина ξ_0 называется *корнем дерева*, из нее исходит пучок из N ребер, образующих 1-*й*

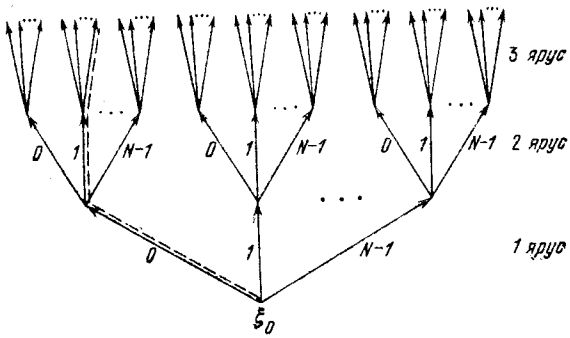


Рис. 2.

ярус. Каждое из ребер 1-го яруса ведет в вершину, из которой в свою очередь исходит пучок из N ребер, образующих 2-*й ярус*, и т. д. Вершины, являющиеся концами ребер m-го яруса, причисляются также к m-му ярусу (вершина ξ_0 считается вершиной 0-го яруса). Ребра каждого пучка нумеруются слева направо числами 0, 1, ..., $N-1$ (см. рис. 2) или их записями в k-ичной системе счисления:

$$\underbrace{(0, \ldots, 0, 0)}_{n}; \underbrace{(0, \ldots, 0, 1)}_{n}; \ldots; \underbrace{(k-1, k-1, \ldots, k-1)}_{n}.$$

В дальнейшем на рисунках номера ребер будут опускаться. Будем называть *ветвью дерева* связное *) подмножество ребер, содержащее в каждом ярусе ровно по одному ребру. Очевидно, что каждой ветви дерева можно сопоставить последовательность

$$\alpha = \{\alpha(1), \alpha(2), \ldots, \alpha(m), \ldots\},$$

где m — номер яруса, а $\alpha(m)$ — номер ребра, входящего в эту ветвь, если идти по ней, начиная от корня. Так, например, ветви, помеченной на рис. 2 штриховой ли-

*) Упорядоченное множество ребер, в котором конец предыдущего ребра является началом последующего.

нией, соответствует последовательность $\{0, 1, N-1, \ldots\}$. Очевидно, что $\alpha \in E_{\mathfrak{c},N}$. Справедливо и обратное утверждение: каждой последовательности α из $E_{\mathfrak{c},N}$ соответствует некоторая ветвь дерева. Таким образом, мы имеем взаимно однозначное соответствие между ветвями дерева и элементами множества $E_{\mathfrak{c},N}$. В силу этого мы можем пользоваться деревом для геометрического задания множества $E_{\mathfrak{c},N}$.

Пусть $f(X)$ является функцией из $P_{\text{д},N}$ $(N = k^n)$ и $X = (x_1, \ldots, x_n)$. Используя соотношение

$$f(X) \sim \{f_1(X_1),\ f_2(X_1, X_2),\ \ldots,\ f_m(X_1, X_2, \ldots, X_m),\ \ldots\},$$

при помощи $f(X)$ каждому ребру дерева припишем число из E_k. Для этого возьмем произвольное ребро из m-го яруса $(m = 1, 2, \ldots)$ и рассмотрим путь, ведущий из корня к этому ребру — он может быть также определен как отрезок произвольной ветви, проходящей через данное ребро. Очевидно, что путь определен однозначным образом и может быть охарактеризован некоторым кортежем $\alpha(1), \alpha(2), \ldots, \alpha(m)$ номеров ребер пути, отсчитываемых от корня. Исходному ребру припишем m-й член выходной последовательности — число $\gamma(m)$, где

$$\gamma(m) = f_m(\alpha(1), \ldots, \alpha(m)).$$

Полученное дерево будем называть *занумерованным деревом* (точнее, *деревом с занумерованными ребрами*).

Пример 3. а) Для функции $f_{\&}(x_1, x_2)$ имеем: $k = n = 2$, $N = 4$ и

$$\gamma(m) = f_m(X_1, X_2, \ldots, X_m) = f_m(X_m) = x_1(m) \,\&\, x_2(m).$$

Следовательно, $\gamma(m)$ зависит только от последнего члена кортежа, ведущего к данному ребру, т. е. от номера ребра.

Ребру с номером $0 = (0, 0)$ соответствует значение $0 \,\&\, 0 = 0$,
« $1 = (0, 1)$ « $0 \,\&\, 1 = 0$,
« $2 = (1, 0)$ « $1 \,\&\, 0 = 0$,
« $3 = (1, 1)$ « $1 \,\&\, 1 = 1$.

На рис. 3 представлено соответствующее занумерованное дерево.

б) Для функции $z = x + y$ имеем $k = 2$, $n = 2$ и $N = 4$. Очевидно, что

$$z(m) = \begin{cases} x(m) + y(m) & (\mathrm{mod}\ 2) \text{ при отсутствии переноса,} \\ x(m) + y(m) + 1 & (\mathrm{mod}\ 2) \text{ при наличии переноса.} \end{cases}$$

Отсюда нетрудно усмотреть закон получения занумерованного дерева (рис. 4). Процесс приписывания ребрам

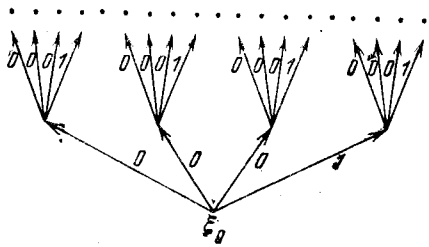


Рис. 3

чисел начинается с 1-го яруса. Затем переходят ко 2-му ярусу и т. д. При этом, если появляется перенос в следующий разряд, то конец соответствующего ребра помечается кружочком. Это позволяет выполнить вычисления в следующем ярусе.

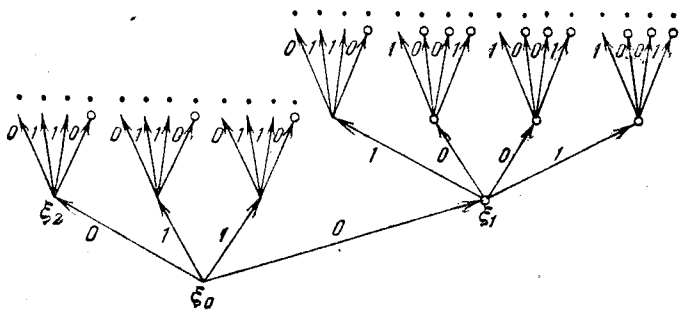


Рис. 4

в) Для функции $z = x^2$ имеем $k = 2$, $n = 1$ и $N = 2$. Здесь явное выражение для $z(m)$ в виде функции $f_m(x(1), \ldots, x(m))$ значительно сложнее, чем в предыдущих случаях, и удобнее вычислять $z(m)$, пользуясь алго-

ритмом возведения в квадрат. Из него видно, что

$$z(1)=x(1),$$
$$z(2)=0,$$
$$z(3)=x(2)+x(1)x(2) \pmod 2,$$
$$z(4)=x(1)x(3)+x(1)x(2) \pmod 2 \text{ и т. д.}$$

Соответственно получаем начальный фрагмент занумерованного дерева (рис. 5).

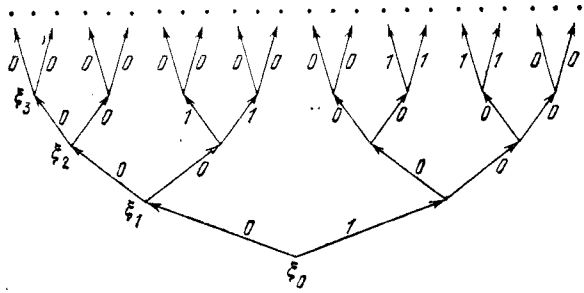


Рис. 5

Итак, мы видим, что по детерминированной функции можно получить занумерованное дерево. Обратное утверждение, вообще говоря, неверно: занумерованное дерево может определять несколько детерминированных функций. Параметры N и k' (где N — число ребер, исходящих из каждой вершины, а k' — максимум чисел, приписанных ребрам) могут допускать несколько решений уравнения $N=k^n$ при $k \geqslant k'$. (Всегда существует решение $k=N$ и $n=1$, т. е. определяется детерминированная функция от одного переменного.) Однако, если по детерминированной функции $f(x_1, \ldots, x_n)$ построить занумерованное дерево, то по этому занумерованному дереву с параметрами k и n определится единственная детерминированная функция, а именно $f(x_1, \ldots, x_n)$. Таким образом, занумерованными деревьями можно пользоваться в качестве аппарата для изучения детерминированных функций.

Возьмем занумерованное дерево для некоторой детерминированной функции $f(X)=f(x_1, \ldots, x_n)$. Пусть ξ — произвольная его вершина m-го яруса. В нее из корня ξ_0 ведет путь

$$\alpha(1), \ldots, \alpha(m)$$

(при $\xi = \xi_0$ путь является пустым). Совокупность всех ветвей, исходящих из ξ, порождает некоторое дерево с корнем ξ, которое будем называть *специальным поддеревом* исходного дерева. Это поддерево определяется множеством всех последовательностей из $E_{t,k}$ с фиксированным началом

$$\alpha(1), \ldots, \alpha(m).$$

Так как исходное дерево занумеровано, то поддерево является также занумерованным. Если в поддереве ввести нумерацию ярусов, начиная с 1-го, то ему будет соответствовать детерминированная функция $f^{\xi}(X)$. Ее можно аналитически определить следующим образом: пусть

$$f(X) \sim \{f_1(X_1), f_2(X_1, X_2), \ldots\},$$
$$f^{\xi}(X) \sim \{f_1^{\xi}(X_1), f_2^{\xi}(X_1, X_2), \ldots\}.$$

Тогда

$$f_i^{\xi}(X_1, \ldots, X_i) = f_{m+i}(\alpha(1), \ldots, \alpha(m), X_1, \ldots, X_i)$$
$$(i = 1, 2, \ldots).$$

Определение. Два поддерева с корнями ξ_1 и ξ_2 исходного дерева называются *эквивалентными*, если

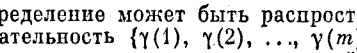
$$f^{\xi_1}(X) \equiv f^{\xi_2}(X).$$

Очевидно, что при естественном наложении двух эквивалентных поддеревьев их нумерации совпадают. Так, в дереве на рис. 3 все поддеревья эквивалентны, а в дереве на рис. 4 поддеревья с корнями ξ_0 и ξ_2 эквивалентны, а с корнями ξ_0 и ξ_1 — не эквивалентны.

Соотношение эквивалентности позволяет в исходном дереве множество всех поддеревьев разбить на классы эквивалентности.

Определение. Число r классов эквивалентности, на которое разбивается множество всех поддеревьев данного дерева, называется *весом дерева* и соответственно *весом детерминированной функции* *).

*) Данное определение может быть распространено и на константы. Последовательность $\{\gamma(1), \gamma(2), \ldots, \gamma(m), \ldots\}$ изображается вырожденным деревом, состоящим из одной ветви



В нем можно рассматривать поддеревья и определить эквивалентность поддеревьев.

Иначе говоря, вес — это максимальное число попарно неэквивалентных поддеревьев. При этом не исключается случай, когда вес r бесконечен.

Обратимся к примеру 3. В дереве для функции $f_{\&}(x_1, x_2)$ (рис. 3), как мы видели, все поддеревья эквивалентны, поэтому $r = 1$. В дереве для функции $z = x + y$ (рис. 4) каждое поддерево эквивалентно либо поддереву с корнем ξ_0, либо поддереву с корнем ξ_1, поэтому $r = 2$.

В дереве для функции $z = x^2$ (рис. 5) поддеревья с корнями $\xi_0, \xi_1, \xi_2, \ldots$ (лежащие на левой ветви) попарно неэквивалентны, так как в силу справедливости соотношения

$$\{\underbrace{0 \ldots 0}_{i}\alpha(i+1), \ldots\}^2 = \{\underbrace{0 \ldots 0}_{2i}\alpha(i+1), \ldots\}$$

поддерево с корнем ξ_i в первых i ярусах заполнено нулями, а в $(i+1)$-м ярусе имеет ребро, которому приписано значение 1. Здесь $r = \infty$.

Для занумерованных деревьев можно ввести нумерацию вершин. Сначала перенумеруем классы эквивалентности числами 0, 1, ... так, чтобы класс, в который по-

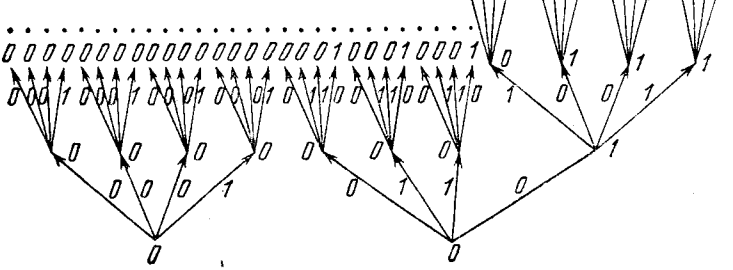


Рис. 6

падает исходное дерево, имел номер 0. Таким образом, нумерация содержит большой произвол. Далее, взяв произвольную вершину ξ, определим класс, в который попадает дерево с корнем ξ. Пусть $\varkappa$ — номер этого класса. Тогда вершине ξ присваивается номер $\varkappa$. Мы получаем дерево, у которого занумерованы также и вершины, причем корень имеет номер 0. На рис. 6 приведены нумера-

ции вершин для двух основных примеров — функций $f_{\&}(x_1, x_2)$ и $z = x + y$.

Если в рассматриваемом дереве сохранить только нумерацию вершин, то легко видеть, что нумерация ребер не восстанавливается однозначным образом. Тем не менее, нумерация вершин весьма полезна при исследовании исходного дерева. В ряде случаев нумерацию вершин можно осуществлять параллельно с нумерацией ребер. Так, в примере для $z = x + y$ номер вершины 0 появляется в случае отсутствия переноса, а 1 — при наличии переноса *). Рассмотрим теперь дерево с занумерованными ребрами и вершинами. Возьмем произвольную ветвь, она проходит через вершины

$$\xi_0, \xi_1, \ldots, \xi_i, \ldots, \xi_j, \ldots$$

Пусть этим вершинам приписаны соответственно номера

$$0, \varkappa_1, \ldots, \varkappa_i, \ldots, \varkappa_j, \ldots$$

Допустим, что $\varkappa_i = \varkappa_j$ $(i \neq j)$ и для всех пар (i, j) $(i \neq j)$, для которых $\varkappa_i = \varkappa_j$, индекс j является наименьшим. Произведем усечение данной ветви, сохранив ее начальный отрезок до вершины ξ_j. Производя эту операцию усечения для каждой ветви, мы получим *усеченное дерево*.

Для случая функции конечного веса r на каждой ветви происходит повторение номеров вершин и номер j,

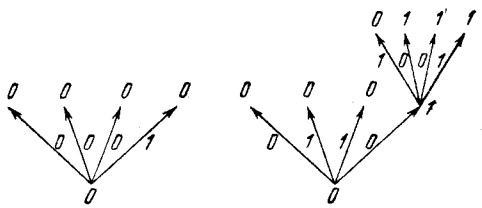


Рис. 7

определяющий усечение, удовлетворяет неравенству $j \leqslant r$. Поэтому для таких функций усеченное дерево будет конечным.

*) Вспомним, что в примере 3, б) при наличии переноса в следующий разряд конец соответствующего ребра мы помечали кружочком.

На рис. 7 приведены усеченные деревья для функций $f_{\&}(x_1, x_2)$ и $z = x + y$. Эти усеченные деревья непосредственно получаются из деревьев, приведенных на рис. 6.

Легко видеть, что усеченное дерево с занумерованными ребрами и вершинами позволяет полностью восстановить исходное занумерованное дерево.

§ 3. Ограниченно-детерминированные функции и способы их задания

Определение. Детерминированная функция $f(x_1, \ldots, x_n)$ называется *ограниченно-детерминированной* (о.-д.) *функцией*, если она имеет конечный вес.

Класс всех о.-д. функций, принадлежащих $P_{д, k}$, обозначим через $P_{од, k}$*).

Примеры 3, а) и 3, б) предыдущего параграфа дают примеры о.-д. функций, а пример 3, в) показывает, что

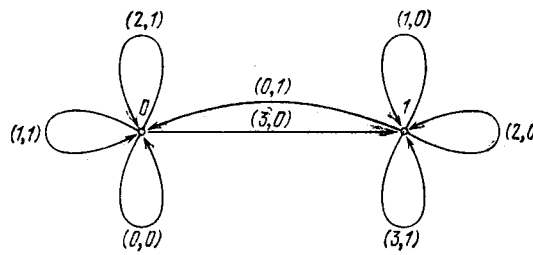


Рис. 8

класс $P_{од, k}$ — класс всех о.-д. функций является собственным подклассом класса $P_{д, k}$ — класса детерминированных функций.

Для любой о.-д. функции соответствующее ей полное (бесконечное) занумерованное дерево можно всегда свести к конечному дереву с занумерованными ребрами и вершинами. Если в этом усеченном дереве произвести отождествление вершин с одинаковыми номерами, то получим так называемую *диаграмму Мура* **). На рис. 8

*) Класс $P_{од, k}$, в частности, содержит все периодические последовательности из $E_{c.k}$.

**) Диаграммы Мура можно использовать и для представления просто детерминированных функций. В этом случае диаграмма будет содержать, вообще говоря, бесконечное число вершин.

приведена диаграмма Мура для функции $z = x + y$. В ней нулем отмечена начальная вершина и для удобства ребрам приписаны пары чисел (α, γ), первое из которых обозначает номер ребра, а второе — число, соответствующее этому ребру.

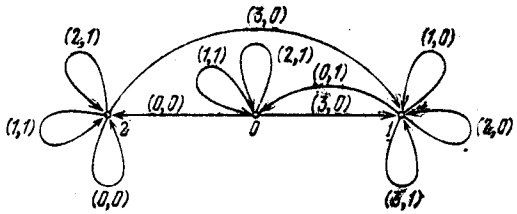


Рис. 9

Таким образом, о.-д. функции можно задавать не только при помощи бесконечных занумерованных деревьев, но и диаграммами Мура. В общем случае, когда f имеет вес r, диаграмма Мура имеет r вершин, причем одна из них выделена в качестве начальной; из каждой вершины исходит $N = k^n$ ребер; ребрам приписаны пары $(0, \gamma')$, $(1, \gamma'')$, ..., $(N-1, \gamma^{(N)})$. В последующем диаграммы, удовлетворяющие данным свойствам, мы будем также называть диаграммами Мура. Диаграммы Мура позволяют строить о.-д. функции любого веса r. При такого рода построениях нужно иметь в виду, что хотя по формально заданной диаграмме Мура о.-д. функция восстанавливается однозначно, однако если по этой о.-д. функции построить диаграмму Мура вышеуказанным способом, то она может не совпасть с исходной. Так, например, диаграмма, представленная на рис. 9, как нетрудно убедиться, задает функцию $z = x + y$ и отлична от диаграммы, приведенной на рис. 8.

Таким образом, не каждая диаграмма Мура с r вершинами изображает о.-д. функцию веса r. Однако диаграммы Мура позволяют оценить число о.-д. функций, зависящих от n переменных $x_1, \ldots, x_n$ и имеющих вес r.

Теорема 2. *Число* $p(k, n, r)$ *о.-д. функций из* $P_{c,k}$, *зависящих от* n *переменных* $x_1, \ldots, x_n$ *и имеющих вес* r, *не превосходит* $(rk)^{rk^n}$.

Доказательство. Возьмем диаграмму Мура для о.-д. функции веса r. В ней из каждой вершины исходит $N = k^n$ ребер, причем α-е ребро соединено с одной из r

вершин и ему приписана пара (α, γ), где $0 \leqslant \gamma \leqslant k-1$. Таким образом, число $p(k, n, r)$ не превосходит числа диаграмм Мура вышеуказанного вида. Данные диаграммы могут быть получены следующим образом.

Возьмем r вершин, занумерованных числами $0, \ldots, r-1$ (0 — выделенная вершина), из каждой из которых исходит по $N = k^n$ ребер, занумерованных числами $0, \ldots, N-1$. Мы имеем rN ребер. Каждое ребро может быть соединено с любой из r вершин и ему может быть приписано любое из k чисел. Поэтому

$$p(k, n, r) \leqslant (rk)^{rN} = (rk)^{rk^n}.$$

Теорема доказана.

Пусть $f(X) = f(x_1, \ldots, x_n)$ — о.-д. функция. Рассмотрим ее диаграмму Мура. Предположим, что в момент $t-1$ мы находились в вершине $\varkappa(t-1)$, тогда при поступлении в момент времени t числа $\alpha(t)$ мы переместимся в диаграмме по ребру $\alpha(t)$, выходящему из вершины $\varkappa(t-1)$ (см. рис. 10), при этом получим выходное значение $\gamma(t)$ и перейдем в вершину $\varkappa(t)$. Таким образом, величины $(\alpha(t), \varkappa(t-1))$ однозначно определяют величины $(\gamma(t), \varkappa(t))$.

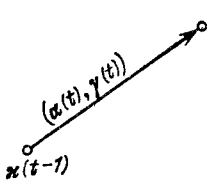


Рис. 10

Введенные ранее величины α и γ будем называть соответственно *входной и выходной величинами*, а $\varkappa$ — *состоянием*.

Пусть переменные X, Q, Z таковы, что: X описывает значение входной величины α, Q описывает значение состояния $\varkappa$ и Z описывает значение выходной величины γ.

На основании приведенных выше рассуждений мы приходим к следующим уравнениям*):

$$Z(t) = \mathcal{F}(X(t), Q(t-1)),$$
$$Q(t) = \mathcal{G}(X(t), Q(t-1)),$$

где $Q(0) = 0$. Данные уравнения называются *каноническими уравнениями*. Нетрудно перейти от векторной за-

*) Для констант из $P_{\text{од}, k}$ аналогичные построения дают уравнения, в которых отсутствует переменная X.

писи канонических уравнений к скалярной. Пусть $l =]\log_k r[$. Тогда

$$z(t) = F'(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(t) = G'_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$
$$q_l(t) = G'_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(0) = \ldots = q_l(0) = 0.$$

Здесь $F', G'_1, \ldots, G'_l$ — функции из P_k, определенные на множестве $\mathscr{E}$, являющемся подмножеством из $\underbrace{E_k \times \ldots \times E_k}_{n+l}$, а именно: переменные $x_1, \ldots, x_n$ пробегают значения из E_k, а вектор $(q_1, \ldots, q_l)$ принимает r значений (например, двоичные записи чисел $0, 1, \ldots, r-1$). Очевидно, что множество $\mathscr{E}$ *является цилиндром* (или *цилиндрическим*) *по* $x_1, \ldots, x_n$ *).

Доопределив функции $F', G'_1, \ldots, G'_l$ на всей области $\underbrace{E_k \times \ldots \times E_k}_{n+l}$, мы получим соответственно функции $F, G_1, \ldots, G_l$ из P_k, и канонические уравнения примут более удобную для нас форму (так как здесь мы можем использовать развитый ранее аппарат формул):

$$z(t) = F(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(t) = G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \qquad (*)$$
$$q_l(t) = G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(0) = \ldots = q_l(0) = 0.$$

Для ограниченно-детерминированных функций, описанных в примерах 3, а) и 3, б), канонические уравнения

---

*) Множество $\mathscr{E} \subset \underbrace{E_k \times \ldots \times E_k}_{n+l}$ называется *цилиндрическим по первым координатам* $x_1, \ldots, x_n$ тогда и только тогда, когда любые два набора $(\zeta'_1, \ldots, \zeta'_n, \eta_1, \ldots, \eta_l)$ и $(\zeta''_1, \ldots, \zeta''_n, \eta_1, \ldots, \eta_l)$ либо одновременно принадлежат $\mathscr{E}$, либо одновременно не принадлежат $\mathscr{E}$.

Очевидно, что область определения $\mathscr{E}$ функций $F', G'_1, \ldots, G'_l$ является цилиндрическим множеством по $x_1, \ldots, x_n$.

имеют, соответственно, следующий вид:

$$z(t) = x_1(t) \& x_2(t);$$

$$z(t) = x(t) + y(t) + q(t-1) \pmod 2,$$

$$q(t) = x(t)y(t) \vee x(t)q(t-1) \vee y(t)q(t-1),$$

$$q(0) = 0.$$

Пример 4. Пусть о.-д. функция $f(x)$ задана диаграммой Мура, приведенной на рис. 11. Для нее функции $\mathcal{F}$ и $\mathcal{G}$ приведены в табл. 1.

Закодировав состояния 0, 1, 2

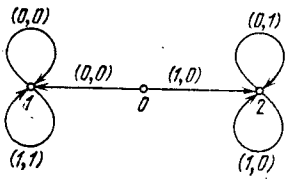


Рис. 11

Таблица 1

X	Q	$\mathcal{F}$	$\mathcal{G}$
0	0	0	1
0	1	0	1
0	2	1	2
1	0	0	2
1	1	1	1
1	2	0	2

наборами (0, 0), (0, 1) и (1, 0), мы получим функции F', G'_1, G'_2 (см. табл. 2).

Таблица 2

x	q_1	q_2	F'	G'_1	G'_2
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	не определены		
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	не определены		

Таблица 3

x	q_1	q_2	F	G_1	G_2
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	1	1	1

Доопределив функции F', G'_1, G'_2, например так, как это сделано в табл. 3, мы получим функции F, G_1 и G_2. Имея теперь функции F, G_1, G_2, получим канонические

уравнения

$$z(t) = \bar{x}(t)\&q_1(t-1) \vee x(t)\&q_2(t-1),$$
$$q_1(t) = q_1(t-1) \vee x(t)\&\bar{q}_2(t-1),$$
$$q_2(t) = q_2(t-1) \vee \bar{x}(t)\&\bar{q}_1(t-1),$$
$$q_1(0) = q_2(0) = 0.$$

Заметим, что в случае, когда $r = 1$, переменные q в канонических уравнениях отсутствуют.

Таким образом каждой о.-д. функции можно сопоставить канонические уравнения. Однако выбор канонических уравнений не однозначен. Эта неоднозначность связана:

а) с различными способами кодирования (нумерации) состояний;

б) с различными способами доопределения функций $F', G'_1, \ldots, G'_l$.

Легко видеть, что канонические уравнения позволяют вычислить по входной последовательности

$$\alpha = \{\alpha(1), \alpha(2), \ldots\}$$

выходную последовательность

$$\gamma = \{\gamma(1), \gamma(2), \ldots\}.$$

Часто рассматривают произвольные системы (*) (у которых l может быть и больше $]\log_k r[$) для задания о.-д. функции. В частности, может оказаться, что если для о.-д. функции f, определяемой этими уравнениями, взять любые канонические уравнения, то они не будут совпадать с исходными уравнениями.

§ 4. Операции над о.-д. функциями

При определении операций над о.-д. функциями удобно исходить из классов $P_{\text{с},k}$ и $P_{\text{д},k}$.

В $P_{\text{с},k}$, так же, как и в P_k вводится операция суперпозиции: сначала определяется понятие формулы над системой функций из $P_{\text{с},k}$, а затем каждой формуле сопоставляется функция из $P_{\text{с},k}$. Легко видеть, что справедлива

Теорема 3. *Класс детерминированных функций замкнут относительно операции суперпозиции.*

Суперпозицию детерминированных функций удобно изображать графически в виде блок-схемы. Если система содержит тождественную функцию, то суперпозиция сводится к многократному применению элементарных суперпозиций вида

$$f(X)=f_0(f_1(X^1),\ \ldots,\ f_m(X^m)).$$

Поэтому достаточно указать, как выглядит блок-схема для элементарной суперпозиции (см. рис. 12). На блок-схеме квадратиками изображены преобразователи, которые реализуют функцию, написанную внутри квадрата.

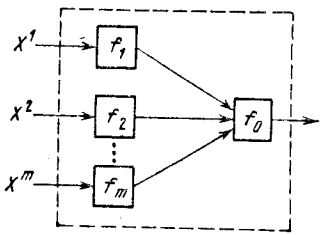


Рис. 12

Теорема 4. *Класс о.-д. функций замкнут относительно суперпозиции.*

Доказательство. Так как класс о.-д. функций содержит тождественную функцию, то для доказательства теоремы достаточно установить, что функция $f(X)$, получаемая из о.-д. функций f_0, f_1, ..., f_m по формуле

$$f(X)=f_0(f_1(X^1),\ \ldots,\ f_m(X^m)),$$

является о.-д. функцией.

Поскольку, как и прежде, функция рассматривается с точностью до несущественных переменных (а при добавлении и изъятии несущественных переменных о.-д. функция переходит в о.-д. функцию с тем же весом), то можно считать, что функции f_1, ..., f_m зависят от одних и тех же переменных x_1, ..., x_n, т. е.

$$f(x_1,\ \ldots,\ x_n)=f_0(f_1(x_1,\ \ldots,\ x_n),\ \ldots,\ f_m(x_1,\ \ldots,\ x_n)).$$

Выпишем канонические уравнения для f_0, f_1, ..., f_m:

$$z_0(t)=F_0(y_1(t),\ \ldots,\ y_m(t),\ q_1^0(t-1),\ \ldots,\ q_{l_0}^0(t-1)),$$

$$q_1^0(t)=G_1^0(y_1(t),\ \ldots,\ y_m(t),\ q_1^0(t-1),\ \ldots,\ q_{l_0}^0(t-1)),$$

. . . . . .

$$q_{l_0}^0(t)=G_{l_0}^0(y_1(t),\ \ldots,\ y_m(t),\ q_1^0(t-1),\ \ldots,\ q_{l_0}^0(t-1)),$$

$$q_1^0(0)=\ldots=q_{l_0}^0(0)=0;$$

$$z_1(t) = F_1(x_1(t), \ldots, x_n(t), q_1^1(t-1), \ldots, q_{l_1}^1(t-1)),$$

$$q_1^1(t) = G_1^1(x_1(t), \ldots, x_n(t), q_1^1(t-1), \ldots, q_{l_1}^1(t-1)),$$

. . . . . .

$$q_{l_1}^1(t) = G_{l_1}^1(x_1(t), \ldots, x_n(t), q_1^1(t-1), \ldots, q_{l_1}^1(t-1)),$$

$$q_1^1(0) = \ldots = q_{l_1}^1(0) = 0;$$

. . . . . .

$$z_m(t) = F_m(x_1(t), \ldots, x_n(t), q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$

$$q_1^m(t) = G_1^m(x_1(t), \ldots, x_n(t), q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$

. . . . . .

$$q_{l_m}^m(t) = G_{l_m}^m(x_1(t), \ldots, x_n(t), q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$

$$q_1^m(0) = \ldots = q_{l_m}^m(0) = 0.$$

(Предполагается, что символы q_j^i для разных пар (i, j) различны.) Покажем, что f можно задать также при помощи канонических уравнений. В самом деле, положим

$$F\left(x_1, \ldots, x_n, q_1^0, \ldots, q_{l_0}^0, q_1^1, \ldots, q_{l_1}^1, \ldots, q_1^m, \ldots, q_{l_m}^m\right) =$$

$$= F_0\left(F_1\left(x_1, \ldots, x_n, q_1^1, \ldots, q_{l_1}^1\right), \ldots\right.$$

$$\left.\ldots, F_m\left(x_1, \ldots, x_n, q_1^m, \ldots, q_{l_m}^m\right), q_1^0, \ldots, q_{l_0}^0\right),$$

$$G_{ij}\left(x_1, \ldots, x_n, q_1^0, \ldots, q_{l_0}^0, q_1^1, \ldots, q_{l_1}^1, \ldots, q_1^m, \ldots, q_{l_m}^m\right) =$$

$$= \begin{cases} G_j^0\left(F_1\left(x_1, \ldots, x_n, q_1^1, \ldots, q_{l_1}^1\right), \ldots\right. \\ \qquad \left.\ldots, F_m\left(x_1, \ldots, x_n, q_1^m, \ldots, q_{l_m}^m\right), q_1^0, \ldots, q_{l_0}^0\right) \\ \qquad\qquad (i = 0, \quad 1 \leqslant j \leqslant l_0), \\ G_j^i\left(x_1, \ldots, x_n, q_1^1, \ldots, q_{l_i}^1\right) \quad (i > 0, \quad 1 \leqslant j \leqslant l_i). \end{cases}$$

Тогда уравнения

$$z(t) = F\left(x_1(t), \ldots, x_n(t), q_1^0(t-1), \ldots, q_{l_0}^0(t-1), \ldots \right.$$
$$\left. \ldots, q_1^m(t-1), \ldots, q_{l_m}^m(t-1)\right),$$

$$q_j^i(t) = G_{ij}\left(x_1(t), \ldots, x_n(t), q_1^0(t-1), \ldots, q_{l_0}^0(t-1), \ldots \right.$$
$$\left. \ldots, q_1^m(t-1), \ldots, q_{l_m}^m(t-1)\right),$$

$$q_j^i(0) = 0,$$

$$0 \leqslant i \leqslant m, \quad 1 \leqslant j \leqslant l_i,$$

задают, очевидно, функцию f. Теорема доказана.

В $P_{\text{д, к}}$ определим операцию O, называемую *операцией введения обратной связи.*

Определение. Детерминированная функция $f(x_1, \ldots, x_i, \ldots, x_n)$ *зависит от переменной* x_i *с запаздыванием*, если для любых входных последовательностей

$$\alpha_1 = \{\alpha_1(1), \alpha_1(2), \ldots, \alpha_1(t), \ldots\},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\alpha_n = \{\alpha_n(1), \alpha_n(2), \ldots, \alpha_n(t), \ldots\}$$

и любого момента времени t значение $\gamma(t)$, где

$$\gamma = f(\alpha_1, \ldots, \alpha_n),$$

полностью определяется значениями первых t членов последовательностей

$$\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$$

и значениями первых $t-1$ членов последовательности α_i (следовательно, $\gamma(t)$ не зависит от $\alpha_i(t)$).

Пример 5. Рассмотрим функцию $f(x)$ из $P_{\text{д, 2}}$, для которой $\gamma(t) = \alpha(t-1)$ и $\gamma(1) = 0$, т. е. $f(x)$ осуществляет сдвиг входной последовательности на один разряд. В дальнейшем мы эту функцию обозначим через $\vec{x}$. На рис. 13 представлено дерево для $\vec{x}$. Из рисунка видно, что $\vec{x}$ — о.-д. функция с весом 2 и имеет следующие канонические уравнения:

$$z(t) = q(t-1),$$

$$q(t) = x(t),$$

$$q(0) = 0.$$

Легко видеть, что функция $\vec{x}$ зависит от x с запаздыванием.

Пусть $f(x_1, \ldots, x_n)$ зависит с запаздыванием от переменных $x_{i_1}, \ldots, x_{i_s}$, тогда, очевидно, для любых входных последовательностей

$$\alpha_1 = \{\alpha_1(1), \alpha_1(2), \ldots, \alpha_1(t), \ldots\},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\alpha_n = \{\alpha_n(1), \alpha_n(2), \ldots, \alpha_n(t), \ldots\}$$

и любого момента t значение $\gamma(t)$, где

$$\gamma = f(\alpha_1, \ldots, \alpha_n),$$

полностью определяется значениями первых t членов последовательностей

$$\alpha_1, \ldots, \alpha_{i_1-1}, \alpha_{i_1+1}, \ldots, \alpha_{i_s-1}, \alpha_{i_s+1}, \ldots, \alpha_n$$

и значениями первых $t-1$ членов последовательностей

$$\alpha_{i_1}, \ldots, \alpha_{i_s}$$

(значит, $\gamma(t)$ не зависит от $\alpha_{i_1}(t), \ldots, \alpha_{i_s}(t)$).

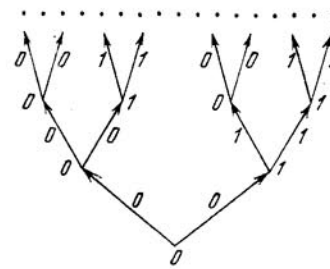


Рис. 13

В случае, когда $f(x_1, \ldots, x_n) \in P_{\text{од. } k}$, можно дать определение зависимости от переменной x_i $(1 \leqslant i \leqslant n)$ с запаздыванием на языке канонических уравнений: $f(x_1, \ldots, x_n)$ зависит от x_i $(1 \leqslant i \leqslant n)$ с запаздыванием тогда и только тогда, когда f можно задать каноническими уравнениями

$$\begin{aligned}
z(t) &= F(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),\\
q_1(t) &= G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (*)\\
q_l(t) &= G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),\\
q_1(0) &= \ldots = q_l(0) = 0,
\end{aligned}$$

в которых функция $F(x_1, \ldots, x_n, q_1, \ldots, q_l)$ как функция из P_k существенно не зависит от x_i.

В примере 5 видно, что $F(x, q) \equiv q$, т. е. F не зависит существенно от x, что согласуется со вторым определением зависимости от переменной с запаздыванием.

Пусть теперь о.-д. функция $f(x_1, \ldots, x_n)$ зависит с запаздыванием от переменных $x_{i_1}, \ldots, x_{i_s}$. Тогда из (*) для каждого x_{i_ν} $(\nu = 1, \ldots, s)$ можно найти канонические уравнения, описывающие f, вида

$$z(t) = F_\nu(x_1(t), \ldots, x_{i_\nu - 1}(t), x_{i_\nu + 1}(t), \ldots$$

$$\ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

$$q_1(t) = G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$q_l(t) = G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

$$q_1(0) = \ldots = q_l(0) = 0,$$

где F_ν получается путем доопределения из некоторой функции F', $F_\nu \in P_k$ и существенно не зависит от x_{i_ν}.

Таблица 4

x_1	x_2	$F'(x_1, x_2)$
0	0	0
1	1	1

Таким образом, из F' мы получаем, вообще говоря, различные доопределения

$$F_1, \ldots, F_s.$$

Возникает вопрос, можно ли подобрать доопределения так, чтобы

$$F_1 \equiv \ldots \equiv F_s \equiv F?$$

(Другими словами — найти такое доопределение F, которое бы существенно не зависело от $x_{i_1}, \ldots, x_{i_s}$.)

Если F' — произвольная не всюду определенная функция из P_k, то гарантировать этого нельзя, в чем можно убедиться на следующем примере.

Пример 6. Пусть $F'(x_1, x_2)$ определена на множестве $\mathscr{E}$, где $\mathscr{E} = \{(0, 0), (1, 1)\}$ и $\mathscr{E} \subset E_2 \times E_2$ (см. табл. 4).

Очевидно, что $F_1(x_1, x_2) \equiv x_1$ и $F_2(x_1, x_2) \equiv x_2$ являются доопределениями F', причем F_1 существенно не зависит от x_2, а F_2 — от x_1. В то же время не существует доопределения F', которое бы существенно не зависело одновременно от x_1 и x_2.

Однако при некоторых ограничениях, которые в нашем случае выполнены, поставленный вопрос допускает положительное решение,

Теорема 5. *Пусть* $F'(x_1, \ldots, x_n, q_1, \ldots, q_l)$ — *функция из* P_k, *определенная на множестве* $\mathscr{E}$, *являющемся цилиндрическим по* $x_1, \ldots, x_n$, *и* F' *допускает доопределения* $F_1, \ldots, F_s$ *такие, что* F_1 *существенно не зависит от* x_{i_1}, F_2 *существенно не зависит от* x_{i_2} *и т. д., наконец,* F_s *существенно не зависит от* x_{i_s}. *Тогда существует такое доопределение* F, *которое существенно не зависит от переменных* $x_{i_1}, \ldots, x_{i_s}$.

Доказательство. Положим

$$F(x_1, \ldots, x_n, q_1, \ldots, q_l) = \begin{cases} F' & \text{на } \mathscr{E}, \\ 0 & \text{вне } \mathscr{E}. \end{cases}$$

Покажем, что F существенно не зависит от $x_{i_1}, \ldots, x_{i_s}$. Пусть $(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l)$ и $(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l)$ — два произвольных набора, которые совпадают для всех переменных, кроме, быть может, $x_{i_1}, \ldots, x_{i_s}$. Тогда в силу цилиндричности $\mathscr{E}$ они оба одновременно либо не принадлежат множеству $\mathscr{E}$, либо оба содержатся в $\mathscr{E}$. В первом случае

$$F(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l) = F(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l) = 0.$$

Во втором случае

$$F(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l) = F'(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l),$$
$$F(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l) = F'(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l).$$

Покажем, что

$$F'(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l) = F'(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l).$$

В самом деле, если это не так, то найдутся два набора $(\zeta_1''', \ldots, \zeta_n''', \eta_1, \ldots, \eta_l)$ и $(\zeta_1^{IV}, \ldots, \zeta_n^{IV}, \eta_1, \ldots, \eta_l)$, соседних по одной из переменных $x_{i_1}, \ldots, x_{i_s}$ (например, x_{i_ν}), для которых из условия $\zeta_i' = \zeta_i''$ следует $\zeta_i''' = \zeta_i^{IV}$ и

$$F'(\zeta_1''', \ldots, \zeta_n''', \eta_1, \ldots, \eta_l) \neq F'(\zeta_1^{IV}, \ldots, \zeta_n^{IV}, \eta_1, \ldots, \eta_l).$$

Но тогда F' невозможно доопределить до функции из P_k, которая существенно бы не зависела от x_{i_ν}, что противоречит исходному допущению.

Таким образом, F существенно не зависит от $x_{i_1}, \ldots, x_{i_s}$ и теорема доказана.

Перейдем теперь к определению операции O.

Пусть $\{f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\}$ $(m \geqslant 2)$ — система детерминированных функций и пусть f_d зависит от переменной x_j с запаздыванием. Тогда, рассматривая эту систему как преобразователь с n входами и m выходами, мы можем d-й выход соединить с j-м входом (см. рис. 14) — ввести «обратную связь» между выходом d и входом j. Мы получим преобразователь, реализующий систему из $m-1$ детерминированных функций

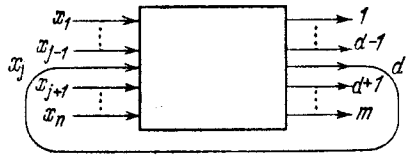


Рис. 14

$$\{f'_1(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n), \ldots$$

$$\ldots, f'_{d-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n),$$

$$f'_{d+1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n), \ldots$$

$$\ldots, f'_m(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)\},$$

зависящих от $n-1$ переменных $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$. Функции $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$ формально определяются так: пусть α' — входная последовательность для $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$.

1) Рассмотрим $\alpha(1) = \{\alpha'_1(1), \ldots, \alpha'_{j-1}(1), 0, \alpha'_{j+1}(1), \ldots$ $\ldots, \alpha'_n(1)\}$. По этому набору вычисляем f_d в момент времени 1, пусть это будет $\gamma_d(1)$.

Рассмотрим $\widetilde{\alpha}(1) = \{\alpha'_1(1), \ldots, \alpha'_{j-1}(1), \gamma_d(1), \alpha'_{j+1}(1), \ldots$ $\ldots, \alpha'_n(1)\}$ и вычислим функции $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$ по этому набору в момент времени 1.

2) Рассмотрим $\alpha(2) = \{\alpha'_1(2), \ldots, \alpha'_{j-1}(2), \gamma_d(1), \alpha'_{j+1}(2), \ldots, \alpha'_n(2)\}$. По наборам $\alpha(1)$ и $\alpha(2)$ (т. е. в конечном счете по наборам $\widetilde{\alpha}(1)$ и $\alpha(2)$ — см. п. 1)) вычисляем f_d в момент времени 2 и получаем $\gamma_d(2)$.

Рассмотрим $\widetilde{\alpha}(2) = \{\alpha'_1(2), \ldots, \alpha'_{j-1}(2), \gamma_d(2), \alpha'_{j+1}(2), \ldots$ $\ldots, \alpha'_n(2)\}$ и по наборам $\widetilde{\alpha}(1)$ и $\widetilde{\alpha}(2)$ вычисляем значения функций $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$ в момент 2 и т. д.

t) Рассмотрим $\alpha(t) = \{\alpha'_1(t), \ldots, \alpha'_{j-1}(t), \gamma_d(t-1), \alpha'_{j+1}(t), \ldots, \alpha'_n(t)\}$. По наборам $\alpha(1), \alpha(2), \ldots, \alpha(t)$ вычисляем f_d в момент времени t и получаем $\gamma_d(t)$.

Рассмотрим $\tilde{\alpha}(t) = \{\alpha'_1(t), \ldots, \alpha'_{j-1}(t), \gamma_d(t), \alpha'_{j+1}(t), \ldots, \alpha'_n(t)\}$, по наборам $\tilde{\alpha}(1), \tilde{\alpha}(2), \ldots, \tilde{\alpha}(t)$ вычисляем значения функций $f_1, \ldots, f_{d-1}, f_{d+1}, \ldots, f_m$ в момент t и т. д.

Если $f_1, \ldots, f_m$ — о.-д. функции, то операция O может быть определена через канонические уравнения. Пусть f_d зависит с запаздыванием от переменной x_j. Возьмем систему канонических уравнений для $f_1, \ldots, f_m$:

$$z_1(t) = F_1(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

. . . . . .

$$z_d(t) = F_d(x_1(t), \ldots, -, \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

. . . . . .

$$z_m(t) = F_m(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

$$q_1(t) = G_1(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

. . . . . .

$$q_l(t) = G_l(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$

$$q_1(0) = \ldots = q_l(0) = 0.$$

Здесь прочерк в наборе аргументов у F_d обозначает, что F_d существенно не зависит от x_j По этой системе канонических уравнений путем выбрасывания d-й строки и заменой переменной x_j на F_d строим новую систему

уравнений:

$$z_1(t) = F_1(x_1(t), \ldots, F_d(x_1(t), \ldots$$
$$\ldots, -, \ldots, x_n(t), q_1(t-1), \ldots$$
$$\ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$z_{d-1}(t) = F_{d-1}(x_1(t), \ldots, F_d(x_1(t), \ldots$$
$$\ldots, -, \ldots, x_n(t), q_1(t-1), \ldots$$
$$\ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$z_{d+1}(t) = F_{d+1}(x_1(t), \ldots, F_d(x_1(t), \ldots$$
$$\ldots, -, \ldots, x_n(t), q_1(t-1), \ldots$$
$$\ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$z_m(t) = F_m(x_1(t), \ldots, F_d(x_1(t), \ldots$$
$$\ldots, -, \ldots, x_n(t), q_1(t-1), \ldots$$
$$\ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(t) = G_1(x_1(t), \ldots, F_d(x_1(t), \ldots$$
$$\ldots, -, \ldots, x_n(t), q_1(t-1), \ldots$$
$$\ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_l(t) = G_l(x_1(t), \ldots, F_d(x_1(t), \ldots$$
$$\ldots, -, \ldots, x_n(t), q_1(t-1), \ldots$$
$$\ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(0) = \ldots = q_l(0) = 0.$$

Эта система уравнений, очевидно, определяет те же о.-д. функции $f_1, \ldots, f_{d-1}, f_{d+1}, \ldots, f_m$, которые были определены выше. Отсюда следует также, что система о.-д. функций, получаемая таким образом, не зависит от выбора канонических уравнений для $f_1, \ldots, f_m$ (при условии, что F_d существенно не зависит от x_j).

Приведем пример, показывающий, каким образом выглядит применение операции O в данном случае. В качестве исходной системы о.-д. функций возьмем систему,

задаваемую каноническими уравнениями

$$z(t) = x(t) + y(t) + q(t-1) \pmod 2,$$
$$w(t) = x(t)y(t) \vee x(t)q(t-1) \vee y(t)q(t-1),$$
$$q(t) = u(t), \quad q(0) = 0.$$

Как было отмечено выше, обе функции z и w зависят от переменной u с запаздыванием. Посредством тождества

$$u(t) = w(t)$$

введем обратную связь. После исключения получим следующие канонические уравнения:

$$z(t) = x(t) + y(t) + q(t-1),$$
$$q(t) = x(t)y(t) \vee x(t)q(t-1) \vee y(t)q(t-1),$$
$$q(0) = 0.$$

Таким образом, результатом операции O является о.-д. функция, представляющая сложение двух последовательностей.

Теорема 6. *Класс о.-д. функций замкнут относительно операции O.*

Введение операции обратной связи можно более компактно охарактеризовать в терминах подстановок функций $F_1, \ldots, F_m$ из P_k, входящих в канонические уравнения.

Рассмотрим случай, когда операция O применяется дважды, и для простоты будем считать, что она применяется к парам (z_1, x_1) и (z_2, x_2) ($(d_1, j_1) = (1, 1)$ и $(d_2, j_2) = (2, 2)$) (что легко достигается перенумерацией функций и переменных). В основе канонических уравнений лежат уравнения

$$z_1 = F_1(x_1, x_2, \ldots, x_n, q_1, \ldots, q_l),$$
$$z_2 = F_2(x_1, x_2, \ldots, x_n, q_1, \ldots, q_l),$$
$$\cdots\cdots\cdots\cdots\cdots$$
$$z_m = F_m(x_1, x_2, \ldots, x_n, q_1, \ldots, q_l),$$

которые, положив $\tilde{u} = (x_3, \ldots, x_n, q_1, \ldots, q_l)$, кратко запишем

$$z_1 = F_1(x_1, x_2, \tilde{u}),$$
$$z_2 = F_2(x_1, x_2, \tilde{u}),$$
$$\cdots\cdots\cdots$$

Так как по условию применима обратная связь (z_1, x_1), то

$$F_1 = F_1(-, x_2, \tilde{u}),$$

т. е. F_1 существенно не зависит от x_1. При помощи подстановки $x_1 = F_1(-, x_2, \tilde{u})$ мы получим систему уравнений

$$z_2 = F_2(F_1(-, x_2, \tilde{u}), x_2, \tilde{u})$$

. . . . . .

По условию возможно введение обратной связи (z_2, x_2). Это означает, что

$$F_2(F_1(-, x_2, \tilde{u}), x_2, \tilde{u})$$

существенно не зависит от x_2. Таким образом, введению обратной связи в последовательности (z_1, x_1), (z_2, x_2) соответствует исключение переменных x_1 и x_2 в исходной системе при помощи подстановок

$$x_1 = F_1(-, F_2(F_1(-, x_2, \tilde{u}), x_2, \tilde{u}), \tilde{u}),$$
$$x_2 = F_2(F_1(-, x_2, \tilde{u}), x_2, \tilde{u}),$$

причем правые части не зависят существенно от x_2.

Пример 7. Рассмотрим систему из трех о.-д. функций

$$z_1(t) = x_2(t)x_3(t) \vee q(t-1),$$
$$z_2(t) = x_1(t)x_2(t) \vee x_3(t) \vee q(t-1),$$
$$z_3(t) = F_3(x_1(t), x_2(t), x_3(t), q(t-1)),$$
$$q(t) = G(x_1(t), x_2(t), x_3(t), q(t-1)),$$
$$q(0) = 0.$$

Здесь можно применить операцию O к (z_1, x_1):

$$F_2(F_1(-, x_2, x_3, q), x_2, x_3, q) =$$
$$= (x_2x_3 \vee q)x_2 \vee x_3 \vee q = x_3 \vee q.$$

Полученная функция существенно не зависит от x_2. Поэтому мы можем далее ввести обратную связь (z_2, x_2). Введению обратной связи в последовательности (z_1, x_1), (z_2, x_2) соответствует пара подстановок

$$x_1 = x_3 \vee q,$$
$$x_2 = x_3 \vee q.$$

С другой стороны, невозможно осуществить введение обратной связи в порядке (z_2, x_2), (z_1, x_1), потому что $F_2(x_1, x_2, x_3, q)$ существенно зависит от переменной x_2.

Данный пример выявляет некоторые негативные стороны введенной нами операции O. Следующая теорема показывает, что для широкого класса ситуаций порядок введения обратных связей безразличен.

Теорема 7. *Пусть для системы о.-д. функций* $\{f_1, f_2, \ldots, f_m\}$, *где* $m \geqslant 3$, *возможно введение обратных связей и в порядке* (z_1, x_1), (z_2, x_2), *и в порядке* (z_2, x_2), (z_1, x_1). *Тогда результаты применения операции* O *совпадают.*

Доказательство. Поскольку к системе о.-д. функций применима операция O как для (z_1, x_1), так и для (z_2, x_2), то

$$F_1 = F_1(-, x_2, \tilde{u}),$$

$$F_2 = F_2(x_1, -, \tilde{u}).$$

В случае применения операции O в порядке (z_1, x_1), (z_2, x_2) мы используем пару подстановок

$$x_1 = F_1(-, F_2(F_1(-, x_2, \tilde{u}), -, \tilde{u}), \tilde{u}),$$

$$x_2 = F_2(F_1(-, x_2, \tilde{u}), -, \tilde{u}),$$

где правые части не зависят существенно от x_2. Аналогично, в случае применения операции O в порядке (z_2, x_2), (z_1, x_1), имеем

$$x_1 = F_1(-, F_2(x_1, -, \tilde{u}), \tilde{u}),$$

$$x_2 = F_2(F_1(-, F_2(x_1, -, \tilde{u}), \tilde{u}), -, \tilde{u}),$$

где правые части не зависят существенно от x_1.

Подставляя в правую часть второго уравнения первой системы вместо x_2 выражение $F_2(x_1, -, \tilde{u})$ и в правую часть первого уравнения второй системы вместо x_1 выражение $F_1(-, x_2, \tilde{u})$, мы получаем тождественные системы. Теорема доказана.

Данная теорема легко обобщается на случай s-кратного применения операции O.

Теорема 8. *Если система о.-д. функций* $\{f_1, \ldots, f_m\}$ *содержит функции* $f_{d_1}, \ldots, f_{d_s}$ $(m \geqslant s+1$; *индексы попарно различны*), *каждая из которых зависит с запаздыванием от переменных* $x_{j_1}, \ldots, x_{j_s}$ (*индексы попарно различны*), *то тогда система функций, получаемая путем*

введения обратных связей $(d_1, j_1), \ldots, (d_s, j_s)$, *не зависит от порядка введения обратных связей.*

Доказательство. В силу того что каждая из функций $f_{d_1}, \ldots, f_{d_s}$ зависит с запаздыванием от всех переменных $x_{j_1}, \ldots, x_{j_s}$, введение обратных связей $(d_1, j_1), \ldots, (d_s, j_s)$ возможно в любом порядке. Тогда на основании обобщенной теоремы 7 получаем, что результаты применения операции O при любых порядках совпадают. Теорема доказана.

Следующий пример показывает, что могут не выполняться условия теоремы 8, а результаты применения операции O не зависят от порядка.

Пример 8. Возьмем систему о.-д. функций

$$z_1(t) = x_2(t)x_3(t) \vee q(t-1),$$
$$z_2(t) = x_1(t) \vee x_3(t) \vee \bar{q}(t-1),$$
$$z_3(t) = F_3(x_1(t), x_2(t), x_3(t), q(t-1)),$$
$$q(t) = G(x_1(t), x_2(t), x_3(t), q(t-1)),$$
$$q(0) = 0.$$

Здесь возможно применение операции O в порядках $(z_1, x_1), (z_2, x_2)$ и $(z_2, x_2), (z_1, x_1)$, и оно определяется одной и той же парой подстановок

$$x_1 = x_3 \vee q,$$
$$x_2 = 1.$$

В дальнейшем при многократном использовании операции O мы, как правило, будем иметь дело с ситуацией, описанной в доказанной теореме.

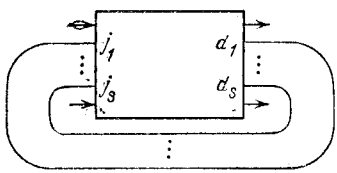


Рис. 15

При тех условиях, для которых справедлива теорема, многократное введение обратных связей $(d_1, j_1), \ldots, (d_s, j_s)$ можно изображать так, как это сделано на рис. 15, ибо в этом изображении нет упорядоченности обратных связей. С другой стороны, многократное введение обратных связей $(d_1, j_1), \ldots, (d_s, j_s)$ для системы о.-д. функций $f_1, \ldots, f_m$ приводит нас к системе из $m - s$ функций от $n - s$ переменных, вычисление которых может быть осуществлено

посредством однократной процедуры типа процедуры на с. 98, т. е. одновременным пересчетом по переменным $x_{j_1}, \ldots, x_{j_s}$*).

Из теорем 4, 6 вытекает

Теорема 9. *Класс $P_{\text{од}, k}$ замкнут относительно операций С и О.*

§ 5. Примеры полных систем

В предыдущем параграфе построена функциональная система $(P_{\text{од}, k}, C, O)$. Теперь мы приступим к рассмотрению вопросов полноты. Для этого необходимо предварительно изучить связь между функциональными системами $(P_{\text{од}, k}, C, O)$ и (P_k, C) и разработать процедуры построения «формул», выражающих о.-д. функции через более простые о.-д. функции.

Как мы видели в § 1, каждой функции Ф из P_k соответствует о.-д. функция f_Φ. Соответствие

$$\Phi \to f_\Phi$$

порождает отображение P_k на некоторое подмножество $\mathscr{P}_k$ о.-д. функций.

Пусть Ф есть суперпозиция функций из P_k, характеризуемая формулой $\mathfrak{A}$, а именно:

$$\Phi = \mathfrak{A}(\Phi_1, \ldots, \Phi_m).$$

---

*) Данное замечание справедливо и в случае, когда $f_1, \ldots, f_m$ — система детерминированных функций, в которой функции $f_{d_1}, \ldots, f_{d_s}$ зависят от переменных $x_{j_1}, \ldots, x_{j_s}$ с запаздыванием.

По аналогии с изображением операции обратной связи для преобразователей введем аналитическую запись для операции обратной связи. Если $f_1, \ldots, f_m$ — система о.-д. функций и $f_{d_1}, \ldots, f_{d_s}$ зависят с запаздыванием от переменных $x_{j_1}, \ldots, x_{j_s}$, то обратные связи $(d_1, j_1), \ldots, (d_s, j_s)$ записываем:

$$\begin{aligned}
z_1 &= f_1(x_1, \ldots, x_n),\\
&\cdots\cdots\cdots\\
x_{j_1} &= f_{d_1}(x_1, \ldots, x_n),\\
&\cdots\cdots\cdots\\
x_{j_s} &= f_{d_s}(x_1, \ldots, x_n),\\
&\cdots\cdots\cdots\\
z_m &= f_m(x_1, \ldots, x_n).
\end{aligned}$$

Рассмотрим о.-д. функции $f_\Phi, f_{\Phi_1}, \ldots, f_{\Phi_m}$, являющиеся образами в этом отображении функций $\Phi, \Phi_1, \ldots, \Phi_m$. Легко видеть, что

$$\mathfrak{A}(f_{\Phi_1}, \ldots, f_{\Phi_m}) = f_{\mathfrak{A}(\Phi_1, \ldots, \Phi_m)} = f_{\Phi},$$

т. е. образ суперпозиции, характеризуемой формулой $\mathfrak{A}$, является суперпозицией образов, характеризуемой той же формулой $\mathfrak{A}$. Это утверждение на языке преобразователей имеет совсем простой смысл. Пусть преобразователь (см. рис. 16) реализует функцию $\Phi(x_1, \ldots, x_m)$ из P_k. Если на входы этого преобразователя подавать значения в моменты времени $t = 1, 2, \ldots$, то он, очевидно, будет реализовывать о.-д. функцию $f_\Phi(x_1, \ldots, x_n)$. Аналогично, пусть блок-схема (см. рис. 17) реализует суперпозицию

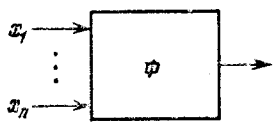


Рис. 16

$$\Phi(x_1, \ldots, x_n) = \Phi_0(\Phi_1(x_1, \ldots, x_n), \ldots, \Phi_m(x_1, \ldots, x_n)),$$

где функции $\Phi_0, \Phi_1, \ldots, \Phi_m$ принадлежат P_k. Если на входы этой блок-схемы подавать значения в моменты

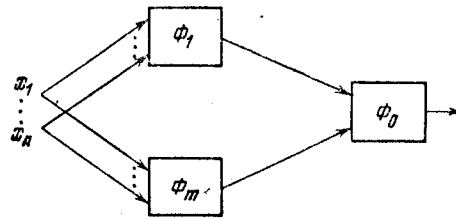


Рис. 17

$t = 1, 2, \ldots$, то она будет реализовывать о.-д. функцию

$$f_{\Phi_0(\Phi_1, \ldots, \Phi_m)} = f_\Phi.$$

Таким образом, отображение P_k на $\mathscr{P}_k$ взаимно однозначно и сохраняет операцию суперпозиции.

Функциональные системы (P_k, C) и $(\mathscr{P}_k, C)$, обладающие указанными свойствами, называются *изоморфными*. Изоморфизм позволяет все результаты для (P_k, C) перенести на $(\mathscr{P}_k, C)$. В частности, отсюда следует, что $\mathscr{P}_k$

имеет конечный базис. В качестве базисов для $\mathscr{P}_k$ можно взять, например:

1) $\{f_0, f_1, \ldots, f_{k-1}, f_{I_0(x)}, \ldots, f_{I_{k-1}(x)}, f_{\min(x_1,x_2)}, f_{\max(x_1,x_2)}\}$, где $f_0, f_1, \ldots, f_{k-1}$ — образы констант 0, 1, ..., $k-1$;

2) $\{f_{V(x_1,x_2)}\}$, где $V(x_1, x_2)$ — функция Вебба.

Для произвольных о.-д. функций важно иметь их представления в терминах операций O и C через более простые о.-д. функции (аналог теоремы о разложении функций в P_2 и P_k, $k \geqslant 3$). Как мы знаем, произвольная о.-д. функция может быть задана при помощи канонических уравнений

$$z(t) = F(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(t) = G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_l(t) = G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(0) = \ldots = q_l(0) = 0.$$

Они «выражают» функцию f с использованием функций из P_k и не являются «формулами» в системе $(P_{\text{од}, k}, C, O)$. Однако нетрудно от них перейти к формуле системы $(P_{\text{од}, k}, C, O)$. Рассмотрим выражение

$$z = f_F(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),$$
$$q_1 = f_{G_1}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_l = f_{G_l}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l).$$

Легко видеть, что оно образовано из функций вида f_Φ (принадлежащих $\mathscr{P}_k$) и $\vec{x}$ путем применения операций C и O: сначала в о.-д. функции $f_F, f_{G_1}, \ldots, f_{G_l}$ на места последних l переменных подставлены о.-д. функции $\vec{q}_1, \ldots, \vec{q}_l$ (операция суперпозиции). Функции

$$f_F(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l), f_{G_1}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l), \ldots$$
$$\ldots, f_{G_l}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l)$$

зависят от переменных $q_1, \ldots, q_l$ с запаздыванием, поэтому возможно введение обратных связей (в силу теоремы 8 они не зависят от порядка) и результат приме-

нения операции O может быть записан в виде данного выражения. Следовательно, это выражение является «формулой» в системе $(P_{\text{од. }k}, C, O)$.

Итак, произвольная о.-д. функция f может быть записана в виде «формулы» (которую мы будем также называть *канонической формулой*) через более простые о.-д. функции при помощи операций C и O.

Теорема 10. *Система о.-д. функций*

$$\{f_0, \ldots, f_{k-1}, f_{I_0}(x), \ldots, f_{I_{k-1}}(x), f_{\min}(x_1, x_2), f_{\max}(x_1, x_2), \vec{x}\}$$

полна в $(P_{\text{од. }k}, C, O)$.

Доказательство. Пусть $f(x_1, \ldots, x_n)$ — произвольная функция из $P_{\text{од. }k}$. Запишем ее в виде канонической формулы

$$z = f_F(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),$$
$$q_1 = f_{G_1}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_l = f_{G_l}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l).$$

Функции $f_F, f_{G_1}, \ldots, f_{G_l}$ принадлежат множеству $\mathcal{P}_k$, порождаются (см. следствие на с. 107) системой

$$\{f_0, \ldots, f_{k-1}, f_{I_0}(x), \ldots, f_{I_{k-1}}(x), f_{\min}(x_1, x_2)\}, f_{\max}(x_1, x_2)\}.$$

Таким образом, каноническая формула может быть записана через функции исходной системы. Теорема доказана.

Аналогично доказывается

Теорема 11. *Система о.-д. функций* $\{f_V(x_1, x_2), \vec{x}\}$ *полна в* $(P_{\text{од. }k}, C, O)$.

Теорема 12. *Пусть* $\{\Phi_1, \ldots, \Phi_m\}$ *— некоторая полная в* (P_k, C) *система. Тогда система о.-д. функций*

$$\{f_{\Phi_1}(x_1, \ldots, x_n), \ldots, f_{\Phi_m}(x_1, \ldots, x_n), \vec{x}\}$$

является полной в $(P_{\text{од. }k}, C, O)$.

Следующее утверждение является обобщением результата В. Б. Кудрявцева [11].

Теорема 13. *Существует о.-д. функция* f *(аналог функции Шеффера) такая, что система* $\{f\}$*, состоящая из одной этой функции, является полной* $(P_{\text{од. }k}, C, O)$.

Доказательство. Пусть $F_0(x_1, x_2, x_3, x_4)$ — функция из P_k, задаваемая формулой

$$F_0(x_1, x_2, x_3, x_4) = \max[x_1 \cdot x_4 + x_3(1 - x_4), x_2] + 1 \pmod{k}$$

(здесь $\cdot$, $+$ и $-$ берутся по $\bmod k$). Рассмотрим о.-д. функцию

$$f(x_1, x_2, x_3, x_4) = f_{F_0}(x_1, x_2, x_3, \vec{x_4}).$$

Покажем, что система $\{f\}$ полна в $(P_{\text{од}, k}, C, O)$. Положим $x_1 = x_3$ и рассмотрим

$$F_0(x_1, x_2, x_1, x_4) = \max[x_1, x_2] + 1 \pmod{k} = V(x_1, x_2),$$

где V — функция Вебба. Тогда в силу упомянутого выше изоморфизма

$$f(x_1, x_2, x_1, x_4) = f_V(x_1, x_2).$$

Данная функция, порождая все $\mathscr{P}_k$, позволяет построить функции f_0, f_1, ..., f_{k-1} и $f_{x+k-1}(x)$. Рассмотрим суперпозицию

$$f_{x+k-1}(f(f_1, f_0, f_0, x_4)) = f_{x+k-1}(f_{F_0}(f_1, f_0, f_0, \vec{x_4})) =$$

$$= f_{F_0+k-1}(f_1, f_0, f_0, \vec{x_4}) = f_{F_0(1,0,0,x_4)+k-1}(\vec{x_4}) = f_{x_4}(\vec{x_4}) = \vec{x_4}.$$

Таким образом, из $\{f\}$ при помощи суперпозиций мы получили $f_V(x_1, x_2)$ и $\vec{x}$, что по теореме 11 дает полноту системы $\{f\}$. Теорема доказана.

Мы видим, что для системы $(P_{\text{од}, k}, C, O)$ сохраняются некоторые свойства, которые справедливы для (P_k, C). Прогноз других свойств системы $(P_{\text{од}, k}, C, O)$ затруднителен, так как, с одной стороны, функциональный объект $P_{\text{од}, k}$ значительно сложнее, чем P_k, и, с другой стороны, добавилась операция O. Первое имеет тенденцию разрушать положительные свойства, второе, наоборот — их усиливать. Заранее сказать трудно, какая тенденция окажется доминирующей. На самом деле в случае проблемы полноты оказалось, что сложность функционального объекта все же влияет сильнее, чем дополнительная операция. Для пояснения приведем без доказательства два результата. Формулировка первого из них может быть понята скорее содержательно, так как в ней используется понятие алгоритма.

Теорема 14 (М. И. Кратко [9, 10]). *Не существует алгоритма, который бы для любой конечной системы о.-д. функций выяснял, является она полной или нет.*

Теорема 15 (В. Б. Кудрявцев [12]). *Мощность множества предполных в* $(P_{\text{од. }k},\ C,\ O)$ *классов равна континууму.*

Таким образом, проблема полноты для $(P_{\text{од. }k},\ C,\ O)$ содержит значительные трудности.

§ 6. О соотношении операций C и O

Функциональная система $(P_{\text{од. }k},\ C,\ O)$ обладает многими специфическими свойствами. Однако их формулировка и изложение требуют значительно больше места, чем, например, для $(P_k,\ C)$. Поэтому здесь мы коснемся только одного вопроса, связанного с тем, что в отличие от рассмотренных, система $(P_{\text{од. }k},\ C,\ O)$ содержит две операции C и O. Речь пойдет о том, в какой мере обе эти операции существенны. Более точно: можно ли выразить результат одной операции через другую и, в частности, отличаются ли друг от друга операции замыкания в системах $(P_{\text{од. }k},\ C)$, $(P_{\text{од. }k},\ O)$ и $(P_{\text{од. }k},\ C,\ O)$?

Сначала докажем некоторые вспомогательные утверждения.

Теорема 16. *Пусть* $X=(x_1, \ldots, x_n)$, *а* $f(X)$ — *о.-д. функция веса* r; *пусть* α — *периодическая последовательность с периодом* p. *Тогда существует* r_1 $(r_1 \leqslant r)$ *такое, что последовательность* γ $(\gamma = f(\alpha))$ *периодическая с периодом* p_1, *где* $p_1 = r_1 p$.

Доказательство. В силу периодичности последовательности α, существует такое t_0, что при $t \geqslant t_0 \geqslant 1$

$$\alpha(t+p)=\alpha(t).$$

Функцию $f(X)$ можно задать диаграммой Мура, содержащей r вершин. Возьмем числовую ось t и в точке t^* $(t^*=1,\ 2,\ \ldots)$ поставим две пометки: числа $\alpha(t^*)$ и $\varkappa(t^*-1)$ — номер вершины, в которую мы попадаем, исходя из начальной вершины и следуя по пути $(\alpha(1), \ldots, \alpha(t^*-1))$ (см. рис. 18). Рассмотрим далее решетку с начальной точкой t_0 и с периодом p (см. рис. 19).

Так как диаграмма Мура содержит r вершин, то в последовательности

$$\varkappa(t_0-1),\ \varkappa(t_0+p-1),\ \ldots,\ \varkappa(t_0+rp-1)$$

по крайней мере два числа совпадают. Пусть i и j таковы, что

$$\varkappa(t_0+ip-1)=\varkappa(t_0+jp-1) \quad (j>i\geqslant 0).$$

Положим $r_1=j-i$. Ясно, что $r_1\leqslant r$. Рассмотрим надрешетку данной решетки с периодом $p_1=r_1p$ и начальной

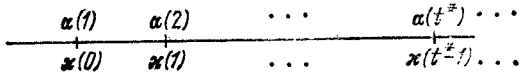


Рис. 18

точкой t_0+ip (см. рис. 20). Очевидно, вершины t_0+ip и t_0+ip+p_1 характеризуются тем, что в них обе пометки α и $\varkappa$ совпадают. Это означает, что в эти моменты времени мы находимся в одной и той же вершине диаграммы

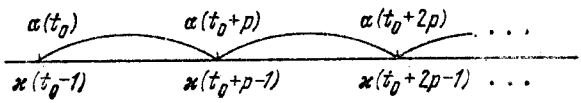


Рис. 19

и перемещаемся затем по одному и тому же ребру. Тогда к следующему моменту мы попадем в одну и ту же вершину, т. е. пометки

$$\varkappa(t_0+ip),\quad \varkappa(t_0+jp),$$

также совпадают. Кроме того, в силу периодичности α

$$\alpha(t_0+ip+1)=\alpha(t_0+jp+1)$$

и т. д. Отсюда вытекает, что и выходные последовательности

$$\{\gamma(t_0+ip),\ \gamma(t_0+ip+1),\ \ldots\},$$
$$\{\gamma(t_0+jp),\ \gamma(t_0+jp+1),\ \ldots\}$$

совпадают, т. е. при $t\geqslant t_0+ip$

$$\gamma(t+p_1)=\gamma(t).$$

Этим теорема доказана.

Теорема 17. *Пусть* $f(X)=f_0(f_1(X),\ \ldots,\ f_m(X))$, *где* $f_0,\ f_1,\ \ldots,\ f_m$ — *о.-д. функции с весами* $r_0,\ r_1,\ \ldots,\ r_m$, *не превосходящими* R_1 *и* α — *периодическая последователь-*

ность, период которой содержит только простые множители, каждый из которых не превосходит R. Тогда $\gamma = f(\alpha)$ — периодическая последовательность с периодом, содержащим только простые множители, каждый из которых не превосходит R.

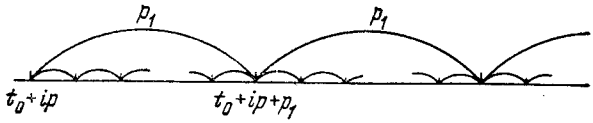


Рис. 20

Доказательство. Пусть α — периодическая последовательность, период которой p содержит только простые множители, не превосходящие R. Рассмотрим последовательности

$$\gamma_1 = f_1(\alpha), \ \ldots, \ \gamma_m = f_m(\alpha).$$

По предыдущей теореме они являются также периодическими с периодами

$$p_1 = r_1' p, \ \ldots, \ p_m = r_m' p \quad (r_1', \ \ldots, r_m' \leqslant R).$$

Очевидно, они содержат также только простые множители, не превосходящие R. Рассмотрим, далее, вектор

$$(\gamma_1, \ \ldots, \ \gamma_m);$$

он будет периодическим и его период p_0 — общее наименьшее кратное периодов $p_1, \ \ldots, \ p_m$; значит, все его простые множители не будут превосходить R. Наконец, последовательность

$$\gamma = f_0(\gamma_1, \ \ldots, \ \gamma_m)$$

по предыдущей теореме будет периодической с периодом p':

$$p' = r_0' p_0 \quad (r_0' \leqslant R).$$

Таким образом, период p' также будет содержать только простые множители, не превосходящие R. Теорема доказана.

На основе этих утверждений мы докажем следующий факт.

Теорема 18. *Система $(P_{\text{од. к}}, C)$ не имеет конечного базиса.*

Доказательство. Пусть имеет место противное, т. е. система $(P_{\text{од. }k}, C)$ обладает базисом $f_1, \ldots, f_s$. Обозначим через $r_1, \ldots, r_s$ веса о.-д. функций $f_1, \ldots, f_s$. Пусть, далее,

$$r = \max(r_1, \ldots, r_s)$$

и p — простое число такое, что $p > r$. Рассмотрим о.-д. функцию $f_\gamma(X)$, которая принимает значение, тождественно равное γ, где γ — периодическая последовательность с периодом p, имеющая вид

$$\gamma = \underbrace{0 \ldots 0}_{p}\,1\,\underbrace{0 \ldots 0}_{p}\,1 \ldots$$

Функцию $f_\gamma(X)$ нельзя выразить при помощи суперпозиции через функции $f_1, \ldots, f_s$. В самом деле, если $f(X)$ — произвольная суперпозиция функций $f_1, \ldots, f_s$, то она будет преобразовывать последовательность $0 = (0, 0, \ldots)$ (ее период равен 1) в последовательность периодическую с периодом, не содержащим простые множители, большие чем r. В то же время $f_\gamma(0) = \gamma$ — периодическая последовательность с периодом $p > r$. Мы пришли к противоречию. Теорема доказана.

Так как система $(P_{\text{од. }k}, C, O)$ имеет конечный базис, а $(P_{\text{од. }k}, C)$ конечного базиса не имеет, то операция O является существенной.

В то же время система $(P_{\text{од. }k}, O)$ по очевидным соображениям не имеет конечного базиса, поэтому операция C является также существенной.

Глава 4

ВЫЧИСЛИМЫЕ ФУНКЦИИ

§ 1. Машины Тьюринга

Из предыдущего следует, что о.-д. функцию $f(x_1, \ldots, x_n)$ можно задать при помощи канонических уравнений

$$Z(t) = F(X(t), Q(t-1)),$$
$$Q(t) = G(X(t), Q(t-1)),$$
$$Q(0) = 0.$$

Эти уравнения позволяют строить по входной последовательности значений переменных $x_1, \ldots, x_n$ выходную последовательность. Более того, значения выходной после-

довательности находятся постепенно по мере поступления входных значений. Это позволяет трактовать выписанные выше уравнения как описание работы некоторого дискретного преобразователя или автомата (рис. 1), который обладает r состояниями (r — вес о.-д. функции) и работает дискретно во времени, формируя состояние памяти и выходное значение в момент времени t по состоянию памяти в момент времени $t-1$ и входным значениям в момент времени t в соответствии с каноническими уравнениями.

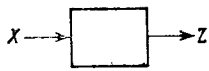


Рис. 1

Данное устройство, в отличие от реальных автоматов (автоматических устройств), никогда не заканчивает работы. Можно ввести другую функциональную характеристику, эквивалентную исходной, но имеющую финитный характер. Из детерминированности функции f вытекает, что отображение f порождает отображение конечных последовательностей вида $\{\alpha(1), \ldots, \alpha(i)\}$ в конечные последовательности $\{\gamma(1), \ldots, \gamma(i)\}$ ($i=1, 2, \ldots$). Это отображение обозначим через $\varphi(x)$. Функцию φ можно задать при помощи тех же канонических уравнений, что и f. Очевидно, что по функции $\varphi(x)$ полностью восстанавливается функция $f(x)$ (в этом смысле выше говорилось об эквивалентности φ и f). Класс функций $\varphi(x)$ обозначим через $\bar{P}_{\text{од. к}}$. Функцию $\varphi(x)$ можно интерпретировать как описание работы автомата в следующем смысле: в моменты времени 1, 2, ..., t на вход поступают символы $\alpha(1)$, $\alpha(2)$, ..., $\alpha(i)$, а на выходе получают символы $\gamma(1)$, $\gamma(2)$, ..., $\gamma(i)$; в последующие моменты времени на вход ничего не поступает и автомат прекращает работу.

В этом случае отсутствие информации на входе равносильно тому, что входной последовательности $\{\alpha(1), \ldots, \alpha(i)\}$ соответствует бесконечная последовательность

$$\{\alpha(1), \ldots, \alpha(i), \Lambda, \ldots\},$$

где Λ — дополнительный символ входного алфавита, означающий отсутствие информации. Появление на входе символа Λ является условием останова устройства, т. е. появления на выходе последовательности

$$\{\gamma(1), \ldots, \gamma(i), \Lambda, \ldots\}.$$

Введенные нами устройства, работающие над конечными входными последовательностями, можно трактовать

также в виде «машины». Она состоит из бесконечной вправо ленты и автомата (см. рис. 2). Бесконечная лента разделена на ячейки, которые нумеруются натуральными числами 1, 2, ...; в ячейки 1, 2, ..., i вписываются символы $\alpha(1)$, $\alpha(2)$, ..., $\alpha(i)$ из алфавита (0, 1, ..., $N-1$). Автомат обладает головкой и может находиться в одном

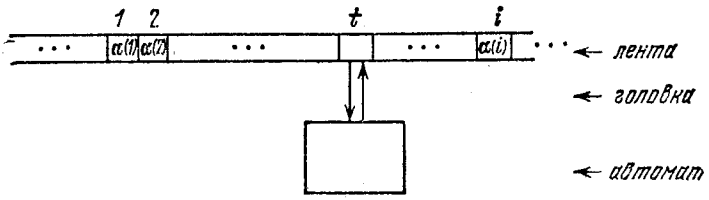


Рис. 2

из (конечного числа) состояний $\varkappa_1$, ..., $\varkappa_r$. Головка в каждый из моментов времени t ($t=1, 2, \ldots$) обозревает одну ячейку ленты (при $t=1$ обозревается 1-я ячейка); по символу, прочитанному на ленте, и по внутреннему состоянию автомат вырабатывает новое состояние и некоторый символ, который через головку вписывает в ту же ячейку (в начальный момент $t=0$ состояние автомата есть $\varkappa_1$). После этого головка сдвигается по ленте на одну ячейку вправо и т. д. Машина останавливается при появлении в поле зрения головки символа Λ, и на ленте в ячейках 1, 2, ..., i получается выходная последовательность $\{\gamma(1), \gamma(2), \ldots, \gamma(i)\}$. Работу этой машины можно задать при помощи так называемой *программы*, т. е. специальной таблицы (см. табл. 1).

В данной таблице строки занумерованы символами Λ, 0, 1, ..., $N-1$, а столбцы — символами $\varkappa_1$, ..., $\varkappa_r$. Строка, соответствующая символу Λ, оставляется незаполненной. В клетку, расположенную в строке α ($\alpha \neq \Lambda$) и в j-м столбце, вписывается тройка символов

$$\{F(\alpha, \varkappa_j), R, G(\alpha, \varkappa_j)\}.$$

Эта тройка называется *командой*. Машина выполняет команду следующим образом: если головка обозревает на ленте символ α и машина к этому моменту находится в состоянии $\varkappa_j$, то в рассматриваемый момент в ячейку вместо символа α записывается символ $F(\alpha, \varkappa_j)$, а машина переходит в состояние $G(\alpha, \varkappa_j)$ и передвигает головку

по ленте на одну ячейку вправо (R). Если читаемым символом является символ Λ, то в соответствующей клетке таблицы стоит «пустая» команда, что рассматривается как команда остановки машины. Очевидно, что программа машины полностью определяется каноническими уравнениями для $\varphi(x)$.

Таблица 1

	x_1	...	x_j	...	x_r
Λ		...		...	
0		...		...	
...	...	...	...	...	...
α		...	$\{F, R, G\}$	...	
...	...	...	...	...	...
$N-1$		...		...	

Данный тип машин можно обобщить, допуская более широкий класс программ и более сложное взаимодействие автомата с лентой.

Так, приведенная на рис. 3 машина (обозначим ее через $\mathfrak{M}$) состоит из бесконечной (но уже в обе стороны) ленты и автомата (см. рис. 3). Ячейки ленты нумеруются целыми числами ..., $-i$, ..., -1, 0, 1, ..., i, ..., в ячейки вписываются символы из алфавита $\{0, 1, \ldots, k-1\}$ (0 в дальнейшем будет играть также роль пустого символа); автомат обладает головкой, способной

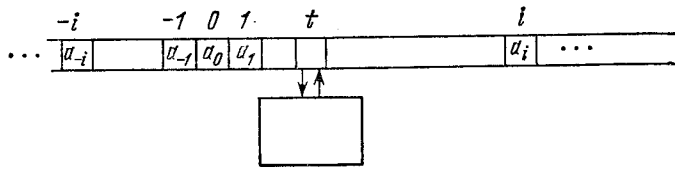


Рис. 3

совершать один из актов: R — сдвигаться на одну ячейку вправо, L — сдвигаться на одну ячейку влево и S — продолжать обозревать ту же ячейку. Символы $\varkappa_1, \ldots, \varkappa_r$ обозначают состояния автомата. Работа машины $\mathfrak{M}$ характеризуется программой T (см. табл. 2). В ней часть клеток может быть незаполненной, т. е. содержать пустые

Таблица 2

	$\varkappa_1$	...	$\varkappa_j$	...	$\varkappa_r$
0		...		...	
...	...	...	...	...	...
a		...	$cD\varkappa$	...	
...	...	...	...	...	...
$k-1$		...		...	

команды, а остальная часть заполняется тройками символов, представляющими команды машины. Например, в клетку, расположенную в строке с номером a и j-м столбце (см. табл. 2), вписана тройка $cD\varkappa$. В команде: c — символ из алфавита $\{0, 1, \ldots, k-1\}$, D — символ из алфавита $\{R, L, S\}$ и $\varkappa$ — одно из состояний $\{\varkappa_1, \ldots, \varkappa_r\}$.

Пусть головка машины обозревает символ a, находясь в состоянии $\varkappa_j$, тогда:

а) если в клетке $(a, \varkappa_j)$ находится команда $(cD\varkappa)$, то машина заменит в этой ячейке символ a на c, перейдет в состояние $\varkappa$, осуществит движение D и приступит к выполнению следующей команды;

б) если в клетке $(a, \varkappa_j)$ стоит пустая команда, машина останавливается.

В начальный момент головка установлена над некоторой ячейкой ленты (начальной ячейкой) и машина находится в состоянии $\varkappa_1$ (соответствующем левому столбцу программы T).

Таким образом, машина, начиная из исходной ситуации (начального состояния и начальной ячейки), осуществляет переработку исходной записи на ленте в соответ-

ствии с программой T. Имеются две возможности: а) при работе машины появится пустая команда — машина останавливается и мы получаем заключительную запись на ленте (состояния, в которых машина останавливается, будем называть *заключительными*); б) при работе машины пустая команда не появится — машина не остановится.

Таблица 3

	x_1
0	
1	$1Rx_1$

Введенные нами машины называются *машинами Тьюринга.*

Пример 1. Пусть $k = 2$. Рассмотрим машину, которая в произвольной записи, начиная из любой ячейки, двигаясь вправо, находит первый нуль. Очевидно, она может быть задана программой, записанной в табл. 3.

В самом деле, возможны три случая.

1) В начальный момент головка видит символ 0. Машина сразу останавливается.

2) В начальный момент головка видит символ 1 и справа от начальной ячейки запись содержит хотя бы один 0. Машина переместит головку через массив из единиц вправо и остановится над первым нулем.

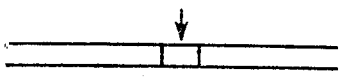


Рис. 4

3) В начальный момент головка видит символ 1 и справа от начальной ячейки запись состоит сплошь из единиц. Машина будет перемещать головку через массив единиц вправо, не останавливаясь.

Теперь введем ряд обозначений и понятий, связанных с записью на ленте. Будем изображать при помощи стрелки положение головки на ленте в рассматриваемый момент времени (см. рис. 4).

То же можно записать еще и так:

$$\ldots a_1 \underset{\downarrow}{a_2} \ldots$$

Здесь стрелка относится к символу a_1, расположенному непосредственно левее стрелки, и обозначает, что головка в данный момент обозревает символ a_1.

Если лента заполнена сплошь нулями, то иногда будем говорить, что мы имеем *пустую ленту.* Для ячейки,

в которой записан символ 0, будет употребляться также термин *пустая ячейка.*

Наконец, совокупность ячеек, которые посетит головка, двигаясь из начальной ячейки до данного момента t, называется *рабочей зоной ленты в момент времени* t.

В дальнейшем нам придется строить машины Тьюринга, обладающие определенными специфическими свойствами. При этом удобно строить машины, исходя из уже построенных машин. Для этого мы введем принцип двойственности и два типа композиции машин.

Таблица 4

	x_1
0	
1	$1Lx_1$

Принцип двойственности для программ (машин). Пусть T — произвольная программа. Обозначим через T^* программу, которая получается из T, если всюду в T заменить в командах R на L и L на R. Программа T^* называется *двойственной к* T.

Пример 2. Программа T^*, задаваемая табл. 4, очевидно, будет двойственной к программе T из предыдущего примера.

Очевидно, что $(T^*)^* = T$, т. е. понятие двойственности программ является взаимным. В последующем мы будем также машины $\mathfrak{M}$ и $\mathfrak{M}^*$, соответствующие программам T и T^*, называть *двойственными машинами.* Легко видеть, что двойственные машины $\mathfrak{M}$ и $\mathfrak{M}^*$ в некотором смысле функционируют симметричным образом, а именно: если в начальный момент на ленте имеется запись

$$\overset{\downarrow}{\ldots a_1\ a_2 \ldots} \tag{1}$$

и машина $\mathfrak{M}$ в момент t ее переработала в запись

$$\overset{\downarrow}{\ldots c_1 c_2 \ldots c_s \ldots}, \tag{2}$$

то машина $\mathfrak{M}^*$ запись

$$\overset{\downarrow}{\ldots a_2 a_1 \ldots}, \tag{3}$$

имеющуюся в начальный момент и симметричную (1)

относительно a_1, перерабатывает в момент t в запись

$$\overset{\downarrow}{\ldots c_s \ldots c_2 c_1 \ldots}_{\mathfrak{z}} \qquad (4)$$

симметричную (2) относительно c_1.

Например, программа T^* (см. пример 2) в силу этого замечания должна, двигаясь влево, отыскивать первый нуль, в чем также можно убедиться непосредственной проверкой.

1-й тип композиции — последовательное подключение одной машины к другой. Пусть $\mathfrak{M}_0$ и $\mathfrak{M}_1$ — две

Таблица 5

	$q_1, \ldots, q_j, \ldots, q_{r_0}$	$q'_1, \ldots, q'_m, \ldots, q'_{r_1}$
0 ... a ... $k-1$		

произвольные машины Тьюринга над одним и тем же входным алфавитом $\{0, 1, \ldots, k-1\}$, множества состояний которых не пересекаются. Перенумеруем числами 0, 1, 2, ..., $l-1$ все пустые клетки (команды) программы T_0 машины $\mathfrak{M}_0$. Пусть $p(x)$ — произвольный предикат*) на множестве $\{0, 1, 2, \ldots, l-1\}$. Построим машину $\mathfrak{M}$, которую и будем называть *последовательным подключением машины* $\mathfrak{M}_1$ *к* $\mathfrak{M}_0$ (относительно предиката $p(x)$). Для этого из таблиц T_0 и T_1 машин $\mathfrak{M}_0$ и $\mathfrak{M}_1$ построим новую таблицу T (см. табл. 5). В ней первая половина совпадает с таблицей T_0 для тех клеток из T_0, в которых стоит непустая команда. В тех клетках η, для которых $p(\eta)=1$, в таблице T стоит команда aSq'_1 (где a — номер строки, в которой находится эта клетка η, а q'_1 — начальное состояние машины $\mathfrak{M}_1$). В тех клетках η, для которых $p(\eta)=0$, в таблице T стоит также пустая команда. Вторая половина таблицы T полностью совпадает с таблицей T_1.

*) Предикат $p(x)$ на $\{0, 1, \ldots, l-1\}$ — специальная функция из P_l. В дальнейшем встречаются *двухзначные* и *трехзначные* предикаты. Они соответственно принимают значения из множеств $\{0, 1\}$ и $\{0, 1, 2\}$.

Очевидно, что работа машины $\mathfrak{M}$ состоит в следующем: исходная запись ленты сначала перерабатывается машиной $\mathfrak{M}_0$, и если машина $\mathfrak{M}_0$ заканчивает работу на команде η такой, что $p(\eta)=1$, то содержимое ленты перерабатывается машиной $\mathfrak{M}_1$. При этом начальной ячейкой для машины $\mathfrak{M}_1$ будет ячейка, в которой остановилась машина $\mathfrak{M}_0$. Таким образом, машина $\mathfrak{M}$ в некотором смысле осуществляет последовательную работу машин $\mathfrak{M}_0$ и $\mathfrak{M}_1$.

2-й тип композиции — итерация машины. Пусть $\mathfrak{M}_0$ — произвольная машина Тьюринга и числами 0, 1, 2, ..., $l-1$ занумерованы пустые клетки ее программы T_0. Пусть $p(x)$ — произвольный предикат на множестве $\{0, 1, 2, \ldots, l-1\}$. Построим машину $\mathfrak{M}$, которую будем называть *итерацией машины* $\mathfrak{M}_0$ *относительно предиката* $p(x)$. Для этого по таблице T_0 построим таблицу T машины $\mathfrak{M}$. Таблица T совпадает с T_0 вне клеток, являющихся пустыми для T_0. В тех клетках η таблицы T_0, для которых $p(\eta)=0$, в таблице T стоит команда $aS\varkappa_1$ (где a — номер строки, в которой находится эта клетка η, а $\varkappa_1$ — начальное состояние машины $\mathfrak{M}_0$). В клетках η таблицы T_0, для которых $p(\eta)=1$, в таблице T стоит также пустая команда.

Легко видеть, что машина $\mathfrak{M}$ в определенном смысле является итерацией машины $\mathfrak{M}_0$, т. е. ее работа эквивалентна многократной работе машины $\mathfrak{M}_0$.

§ 2. Один метод построения машин Тьюринга

Здесь будет описан способ построения машин Тьюринга, который использует композиции машин и специальный операторный язык для записи алгоритмов. Этот язык впервые был предложен А. А. Ляпуновым в 1953 г. (см. [21]). Поскольку сам язык носит вспомогательный характер, то не имеет смысла давать его строгое формально-логическое определение. Мы остановимся лишь на кратком описании операторного языка и рассмотрим серию примеров.

1. Исходными объектами являются операторы, которые подразделяются на три группы:

а) операторы, осуществляющие преобразование записи ленты, состояний машины и перемещение головки машины. Эти операторы обозначаются заглавными латинскими буквами A, B, ... (иногда снабженными индексами);

б) операторы проверки логических условий, обозначаемые символами $p\uparrow$ или $p\updownarrow$ и иногда снабженные индексами;

в) специальные операторы, обозначаемые символами * и ω.

2. Из операторов по определенным правилам строятся операторные схемы. Операторная схема представляет собой некоторую последовательность операторов, в которой для всех стрелок в операторах проверки логических условий указаны операторы, к которым эти стрелки ведут. Например, выражение

$$*A_0 p \overset{\frown}{A_1} \omega$$

будет операторной схемой.

3. Каждой операторной схеме сопоставляется некоторый алгоритм, характеризующий преобразование записи ленты, состояний машины и перемещение головки машины. Последние осуществляются при помощи следующих правил.

а) Операторы в схеме «работают» в определенной последовательности. В данный момент начинает работать тот оператор, перед которым стоит символ *.

б) Пусть мы имеем $*A$. Тогда запись на ленте, состояние машины и положение головки на ленте, имеющиеся к данному моменту, преобразуются оператором A в некоторую запись на ленте, некоторое состояние машины и некоторое положение головки. После этого фрагмент схемы $*A$ перейдет в фрагмент $A*$, что означает, что преобразуется также и операторная схема.

в) Пусть мы имеем $*p\uparrow$ (или $*p\updownarrow$). В этом случае происходит вычисление предиката p по имеющейся записи на ленте и состоянию машины. В случае, если $p=1$, то фрагмент $*p\uparrow$ преобразуется в $p\uparrow*$, т. е. мы перейдем к выполнению следующего оператора; если $p=0$, то фрагмент $*p\uparrow$ преобразуется в $p\overset{*}{\uparrow}$, т. е. выполняется далее оператор, к которому ведет стрелка.

Для $*p\updownarrow$ — аналогичное правило, здесь p — трехзначный предикат, и в зависимости от его значения по соглашению происходит одна из трансформаций

$$*p\updownarrow \Rightarrow p\updownarrow *_1,\quad *p\updownarrow \Rightarrow p\overset{*}{\updownarrow},\quad *p\updownarrow \Rightarrow p\overset{*}{\updownarrow}.$$

г) Сочетание $*\omega$ обозначает конец преобразований или окончание работы.

Пример 3. Рассмотрим операторную схему

$$*A_0 p A_1 \omega.$$

Пусть операторы A_0 и A_1 обозначают преобразования записи ленты, состояний и положения головки, осуществляемые соответственно машинами $\mathfrak{M}_0$ и $\mathfrak{M}_1$. Предположим, что состояния машин $\mathfrak{M}_0$ и $\mathfrak{M}_1$ не пересекаются. Пусть, наконец, $p = p(x)$ — предикат, определенный на номерах ячеек (т. е. совокупности пар $(a, \varkappa)$), на которых машина $\mathfrak{M}_0$ останавливается. Тогда операторная схема осуществляет следующее преобразование.

1. Имеем $*A_0$. Работает машина $\mathfrak{M}_0$ до останова (если он происходит) в ячейке с номером, скажем, η. Схема перейдет в

$$A_0 * p A_1 \omega.$$

2. Имеем $*p$. Вычисляется предикат p и:

а) если $p = 1$, то получаем схему

$$A_0 p * A_1 \omega;$$

б) если $p = 0$, то получаем схему

$$A_0 p A_1 * \omega.$$

3. а) Имеем $*A_1$. Работает машина $\mathfrak{M}_1$. В случае останова машины $\mathfrak{M}_1$, приходим к схеме

$$A_0 p A_1 * \omega.$$

б) Имеем $*\omega$. Преобразование закончено. Преобразование, осуществляемое данной операторной схемой, является преобразованием, осуществляемым последовательным подключением машины $\mathfrak{M}_1$ к $\mathfrak{M}_0$ относительно предиката $p(x)$.

Пример 4. Легко видеть, что операторная схема

$$*A_0 p \omega$$

может интерпретироваться как схема, задающая преобразование, которое получается при итерации машины относительно предиката p.

Теперь перейдем к описанию технологии программирования для машин Тьюринга, т. е. к описанию метода построения программы машины Тьюринга, осуществляющей заданное преобразование. Этот метод параллельно будет иллюстрироваться одним примером. Весь процесс программирования разбивается на четыре этапа.

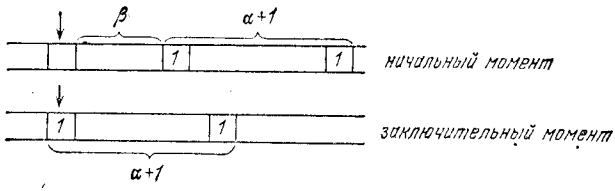


Рис. 5

I этап. Пусть задано некоторое преобразование записи на ленте и положения головки. Допустим, что существует машина Тьюринга, осуществляющая это преобразование. Сначала из неформальных соображений составляют план осуществления данного преобразования. При этом стараются данное преобразование расчленить на более «простые преобразования», т. е. такие, для которых либо мы уже ранее построили машину Тьюринга, либо ее легко можно построить.

Пример 5. Пусть $k=2$. Предположим, что требуется построить машину Тьюринга, которая осуществляет сдвиг массива из $\alpha+1$ единиц ($\alpha=0, 1, \ldots$) влево на $\beta+1$ ячеек ($\beta=1, 2, \ldots$) (вне массива лента пустая). Величину сдвига $\beta+1$ можно задать путем специальной установки головки в начальный момент. На рис. 5 изображены записи на ленте и положение головки в начальный и в заключительный моменты. Требуемое преобразование можно осуществить так:

1) Отмечаем начальную ячейку путем замены в ней символа 0 на символ 1. Оператор для этого преобразования обозначим через $\Phi(0^{\downarrow}\to 1^{\downarrow})$. В скобках показано, что символ 0 заменяется на 1 и головка остается на месте.

2) Движемся вправо (оператор A_1) до левой единицы массива.

3) Обозначим через a' и a'' соответственно символ, обозреваемый в данный момент, и символ, расположенный непосредственно справа от a'. Здесь $a' = 1$. Выясняем, является ли единица, т. е. a', последней единицей в массиве. Для этого вычисляем предикат $p(a', a'')$, где

$$p(a', a'') = \begin{cases} 0 & \text{при } a'' = 0, \\ 1 & \text{при } a'' = 1, \end{cases}$$

В случае $p = 1$

4) Стираем a' (оператор $0(a')$).

5) Возвращаемся влево до ближайшей единицы (оператор A_2).

6) Приписываем справа от этой единицы еще одну единицу. Обозначим через $\Phi(1^{\downarrow} \to 11^{\downarrow})$ соответствующее преобразование.

7) Возвращаемся (оператор p_0, p_0 — предикат, равный тождественно нулю) к A_1.

В случае $p = 0$

8) Массив состоит из одной единицы. Производим $0(a')$ — стирание единицы.

9) Возвращаемся влево (оператор A_2) до ближайшей единицы.

10) Движемся влево (оператор L) через массив из единиц и останавливаемся над левой единицей.

II этап. Переход от составленного плана преобразования к операторной схеме.

Продолжение примера 5. В нашем случае операторная схема имеет вид

$$*\Phi(0^{\downarrow} \to 1^{\downarrow})\, \underline{\uparrow A_1 p(a'a'') \overline{|0(a')\, A_2 \Phi(1^{\downarrow} \to 11^{\downarrow})\, p_0|}}\, {}^{\downarrow} 0(a')\, A_2 L \omega.$$

Замечание. При составлении операторной схемы важно, чтобы заключительное положение головки после выполнения очередного оператора совпадало с начальным положением следующего оператора. Иногда для этого приходится вводить дополнительные согласующие операторы.

В нашем примере согласование получается естественным образом, кроме оператора $p(a'a'')$, для которого мы считаем, что после его выполнения головка встает над a'.

III этап. Переход от операторной схемы к программе машины.

Сначала составляют программы для каждого из операторов данной операторной схемы (кроме операторов $*$, p_0, ω).

Продолжение примера 5. Мы имеем следующие программы для операторов, входящих в схему (если операторы встречаются в схеме несколько раз, то программу составляем для одного из них; см. табл. 6).

Таблица 6

$\Phi(0^{\downarrow}-1^{\downarrow})$	$\varkappa_1$
0	$1S\varkappa_1$
1	

A_1	$\varkappa_2$	$\varkappa_3$
0		$0R\varkappa_3$
1	$1R\varkappa_3$	

$p(a'a'')$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$
0		$0L\varkappa_7$		
1	$1R\varkappa_5$	$1L\varkappa_6$		

$0(a')$	$\varkappa_8$
0	
1	$0S\varkappa_8$

A_2	$\varkappa_9$
0	$0L\varkappa_9$
1	

$\Phi(1^{\downarrow}-11^{\downarrow})$	$\varkappa_{10}$	$\varkappa_{11}$
0	$1S\varkappa_{11}$	
1	$1R\varkappa_{10}$	

L	$\varkappa'$	$\varkappa''$
0	$0R\varkappa''$	
1	$1L\varkappa'$	

При написании программ для операторов состояния выбираем так, чтобы множества состояний для разных операторов не пересекались.

Далее, операторную схему можно рассматривать как схему, определяющую композицию программ (машин). Для этого сначала «оборвем» все стрелки (кроме стрелок типа $^{\downarrow}$). Будем считать, что каждый конец стрелки ведет к своему состоянию останова.

В примере 5 получим следующую схему:

$$*\Phi(0^{\downarrow}\to 1^{\downarrow})A_1p(a'a'')\,{}^{\uparrow}0(a')A_2\Phi(1^{\downarrow}\to 11^{\downarrow})p_0{}^{\downarrow}0(a')A_2L\omega.$$

Затем, беря фрагмент этой схемы

$$A_1p(a'a'')\,{}^{\uparrow}0(a')A_2\Phi(1^{\downarrow}\to 11^{\downarrow}),$$

строим программу, являющуюся последовательным подключением соответствующих программ. После этого переходим к фрагменту

$$\underline{\uparrow A_1p(a'a'')\,{}^{\uparrow}0(a')A_2\Phi(1^{\downarrow}\to 11^{\downarrow})p_0\downarrow}$$

и строим программу, применяя операцию итерации к предыдущей программе относительно предиката p_0. Далее, берем фрагмент

$$*\Phi(0^{\downarrow}\to 1^{\downarrow})\underline{\uparrow A_1p(a'a'')\,{}^{\uparrow}0(a')A_2\Phi(1^{\downarrow}\to 11^{\downarrow})p_0\downarrow}$$

и по нему строим программу, беря последовательное подключение предыдущей программы к программе для $\Phi(0^{\downarrow} \to 1^{\downarrow})$. И, наконец, вся схема

$$* \Phi\left(0^{\downarrow} \to 1^{\downarrow}\right) \uparrow A_1 p(a'a'') \overline{| 0(a') A_2 \Phi\left(1^{\downarrow} \to 11^{\downarrow}\right) p_0 |} 0(a') A_2 L\omega$$

приводит к программе, являющейся последовательным подключением построенной программы и программ для

Таблица 7

	$\Phi(0^{\downarrow}-1^{\downarrow})$		A_1		$p(a'a'')$			$0(a')$
	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$	$\varkappa_8$
0	$1S\varkappa_1$		$0R\varkappa_3$		$0L\varkappa_7$	$0S\varkappa_8$	$0S\varkappa_{12}$	$0S\varkappa_9$
1	$1S\varkappa_2$	$1R\varkappa_3$	$1S\varkappa_4$	$1R\varkappa_5$	$1L\varkappa_6$	$1S\varkappa_8$	$1S\varkappa_{12}$	$0S\varkappa_8$

	A_2	$\Phi(1^{\downarrow}-11^{\downarrow})$		$0(a')$	A_2	L	ω
	$\varkappa_9$	$\varkappa_{10}$	$\varkappa_{11}$	$\varkappa_{12}$	$\varkappa_{13}$	$\varkappa_{14}$	$\varkappa_{15}$
0	$0L\varkappa_9$	$1S\varkappa_{11}$	$0S\varkappa_2$	$0S\varkappa_{13}$	$0L\varkappa_{13}$	$0R\varkappa_{15}$	
1	$1S\varkappa_{10}$	$1R\varkappa_{10}$	$1S\varkappa_2$	$0S\varkappa_{12}$	$1S\varkappa_{14}$	$1L\varkappa_{14}$	

$0(a')$, A_2 и L. Окончательный вид этой программы дан в табл. 7.

Замечание. При последовательном подключении программ, соответствующих двум соседним операторам B' и B'' операторной схемы, образующим фрагмент $B'B''$, предикат p полагается равным тождественно 1, если оператор B' является оператором, преобразующим запись ленты, состояния машины и положения головки. Если B' есть оператор проверки логического условия, то выбор предиката p согласуется с этим логическим условием.

IV этап. Упрощение программы. Построенные данным методом программы иногда допускают значительное упрощение по числу состояний. Здесь мы сформулируем некоторые принципы упрощений.

Допустим, что программа имеет два состояния $\varkappa'$ и $\varkappa''$ таких, что соответствующие им столбцы имеют пустые

клетки, в которые машина никогда не попадает. Пусть для каждой строки существует пустая клетка данного вида (по крайней мере в одном из указанных столбцов). Легко видеть, что тогда можно состояния $\varkappa'$ и $\varkappa''$ отождествить.

Другой тип упрощений связан с операторами проверки логических условий. Мы поясним его на нашем примере.

Продолжение примера 5. Состояния $\varkappa_6$ и $\varkappa_7$ связаны с командами, которые осуществляют только переходы к другим состояниям. Их можно исключить, скорректировав команды для $\varkappa_5$.

Таблица 8

	$\varkappa_1$
0 1	$1S\varkappa_2$

Еще одна возможность упрощений связана с командами, содержащими символ движения S. Пусть в клетке $(c, \varkappa)$ находится команда $c'D\varkappa'$. Тогда, если к этой команде можно непосредственно перейти только от команд $cS\varkappa$, и она не работает в начальный момент, то команду $c'D\varkappa'$ в клетке $(c, \varkappa)$ можно изъять, а все команды $cS\varkappa$ заменить на $c'D\varkappa'$.

Продолжение примера 5. В программе из табл. 7 к команде $1S\varkappa_2$, находящейся в клетке $(1, \varkappa_1)$, можно непосредственно попасть только из команды $1S\varkappa_1$

Таблица 9

	$\varkappa_1$	$\varkappa_3$	$\varkappa_5$	$\varkappa_8$	$\varkappa_9$	$\varkappa_{10}$	$\varkappa_{12}$	$\varkappa_{13}$	$\varkappa_{14}$	$\varkappa_{15}$
0	$1S\varkappa_1$	$0R\varkappa_3$	$0L\varkappa_{12}$	$0S\varkappa_9$	$0L\varkappa_9$	$1S\varkappa_1$	$0S\varkappa_{13}$	$0L\varkappa_{13}$	$0R\varkappa_{15}$	
1	$1R\varkappa_3$	$1R\varkappa_5$	$1L\varkappa_8$	$0S\varkappa_8$	$1S\varkappa_{10}$	$1R\varkappa_{10}$	$0S\varkappa_{10}$	$1S\varkappa_{14}$	$1L\varkappa_{14}$	

того же столбца. Поэтому первый столбец программы может быть заменен на следующий (см. табл. 8). После этого можно отождествить состояния $\varkappa_1$ и $\varkappa_2$, так как в ячейках $(1, \varkappa_1)$ и $(0, \varkappa_2)$ стоят пустые команды и в них машина никогда не попадает.

Аналогичное преобразование можно проделать со столбцами для $\varkappa_3$ и $\varkappa_4$: изъять команду $1R\varkappa_5$ и отождествить состояния $\varkappa_3$ и $\varkappa_4$. Наконец, в столбце $\varkappa_{11}$ команда $0S\varkappa_2$ никогда не работает и ее можно изъять, а $1S\varkappa_2$ можно

также изъять, внеся необходимые изменения в столбец для $\varkappa_{10}$.

Мы приходим к программе (см. табл. 9), содержащей 10 состояний.

§ 3. Машинные коды и их преобразования

В дальнейшем мы будем рассматривать только машины, у которых входной алфавит состоит из двух символов.

Работа машины Тьюринга зависит от характера исходной записи на ленте. Далее чаще будут употребляться специальные виды этих записей, называемые *машинными кодами*. Здесь мы различаем два типа кодов: основные и вспомогательные.

Основные машинные коды имеют следующий вид:

$$\ldots 0\underbrace{1\ldots1}_{\alpha+1}0\ldots$$

— массив из $\alpha+1$ единиц;

$$\ldots 0\underbrace{1\ldots1}_{\alpha_1+1}0\underbrace{1\ldots1}_{\alpha_2+1}0\ldots0\underbrace{1\ldots1}_{\alpha_s+1}0\ldots$$

— s массивов из α_1+1, α_2+1, ..., α_s+1 единиц соответственно, разделенных одним нулем.

Основные машинные коды предназначены для задания чисел α и наборов чисел α_1, α_2, ..., α_s из расширенного натурального ряда (множество, содержащее натуральные числа и нуль). Здесь кодом нуля является запись на ленте, имеющая ровно одну единицу.

С основными кодами связан ряд задач. Мы рассмотрим одну из них, относящуюся к нахождению левой единицы в основном коде. Более точно: требуется построить машину, которая для любого основного кода и любого начального положения головки преобразует основной код в себя (оставляет его на том же месте) и встает над левой единицей кода.

Дадим подробное решение этой задачи.

I этап. План работы искомой машины. Пусть исходная запись на ленте имеет вид $\ldots a_0 \overset{1}{a_1} a_2 \ldots$

1) Выясняем, не пуста ли начальная ячейка, т. е. проверяем условие $p(a_1 \neq 0)$.

2) Пусть $p(a_1 \neq 0) = 1$, т. е. в начальный момент головка обозревает символ 1. Тогда отыскиваем левый конец (т. е. левую единицу) основного кода (оператор A_1) и останавливаемся.

3) Пусть $p(a_1 \neq 0) = 0$. Тогда выясняем, будет ли пустой ячейка, расположенная непосредственно слева от a_1, т. е. проверяем условие $p(a_0 = 0)$.

4) Если $p(a_0 = 0) = 0$, то возвращаемся к выполнению оператора A_1.

5) Если $p(a_0 = 0) = 1$, т. е. $a_0 = 0$, то символы a_0, a_1 заменяем на две единицы и останавливаемся над левой из них (оператор $\Phi(a_0^{\downarrow} a_1 \to 1^{\downarrow}1)$). Таким образом, вне основного кода построен сегмент, концами которого являются единицы. Первоначально длина его равна двум, но в дальнейшем мы будем его увеличивать путем смещения левой единицы влево, а правой — вправо.

6) Выясняем возможность смещения левой единицы сегмента влево на одну ячейку путем проверки условия $p_л$, где

$$p_л = \begin{cases} 1, \text{ если непосредственно слева от сегмента} \\ \qquad\qquad \text{находится пустая ячейка,} \\ 0 \text{ в противном случае.} \end{cases}$$

Пусть $p_л = 1$. Тогда:

7) Осуществляем смещение левой единицы на одну ячейку влево и затем движемся вправо до правой единицы сегмента (оператор A_2).

8) Выясняем возможность смещения правой единицы сегмента на одну ячейку вправо путем проверки условия $p_п$, где

$$p_п = \begin{cases} 1, \text{ если непосредственно справа от сегмента} \\ \qquad\qquad \text{находится пустая ячейка,} \\ 0 \text{ в противном случае.} \end{cases}$$

Если $p_п = 1$, то

9) Осуществляем смещение правой единицы на одну ячейку вправо и затем движение влево до левой единицы (оператор A_3) и возвращаемся к 6).

Если $p_л = 0$, то

(10) Левая единица касается массива из единиц. Идем направо и стираем обе единицы сегмента. После этого возвращаемся влево до правого конца массива (оператор A_4) и затем переходим к A_1.

Если $p_п = 0$, то

11) Правая единица касается массива из единиц. Идем влево, стираем обе единицы сегмента. После этого возвращаемся вправо до левого конца массива (оператор A_5) и затем останавливаемся.

Таблица 10

$p(a_1 \neq 0)$	$\varkappa_1$
0	
1	

A_1	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$
0	$0L\varkappa_3$	$0R\varkappa_4$	$0R\varkappa_4$
1	$1L\varkappa_2$	$1L\varkappa_2$	

$p(a_0 = 0)$	$\varkappa_5$	$\varkappa_6$
0	$0L\varkappa_6$	
1		

Ф	$\varkappa_7$	$\varkappa_8$
0	$1R\varkappa_8$	$1L\varkappa_8$
1		

$p_л$	$\varkappa$	$\varkappa_{10}$
0		
1	$1L\varkappa_{10}$	

A_2	$\varkappa_{11}$	$\varkappa_{12}$
0	$1R\varkappa_{11}$	$0R\varkappa_{12}$
1	$0R\varkappa_{12}$	

$p_п$	$\varkappa_{13}$	$\varkappa_{14}$
0		
1	$1R\varkappa_{14}$	

A_3	$\varkappa_{15}$	$\varkappa_{16}$
0	$1L\varkappa_{15}$	$0L\varkappa_{16}$
1	$0L\varkappa_{16}$	

A_4	$\varkappa_{17}$	$\varkappa_{18}$	$\varkappa_{19}$	$\varkappa_{20}$
0	$1R\varkappa_{18}$	$0R\varkappa_{19}$	$0R\varkappa_{19}$	$0L\varkappa_{20}$
1			$0L\varkappa_{20}$	

A_5	$\varkappa_{21}$	$\varkappa_{22}$	$\varkappa_{23}$	$\varkappa_{24}$
0	$1L\varkappa_{22}$	$0L\varkappa_{23}$	$0L\varkappa_{23}$	$0R\varkappa_{21}$
1			$0R\varkappa_{24}$	

II этап. Запись операторной схемы. В нашем случае она имеет вид

$$*p(a_1 \neq 0)\uparrow\overbrace{A_1\omega}^{\downarrow} p\,(a_0 = 0)\,|\,\Phi\,(a_0 \downarrow a_1 \to 1 \downarrow 1)\overbrace{p_л|\,A_2 p_п|\,A_3 p_0\uparrow A_4 p_0}^{\downarrow}|\uparrow A_5\omega$$

III этап. Составление программ для отдельных операторов (см. табл. 10). Здесь программы для $p_п$, A_3, A_5 двойственны соответственно $p_л$, A_2, A_4.

По операторной схеме и программам для операторов составляем программу рассматриваемой задачи (см. табл. 11).

IV этап. Производим упрощение программы. Команды, расположенные в столбцах x_5, x_7, x_9, x_{13}, x_{17}, x_{21},

Таблица 11

	$p(a_1 \neq 0)$		A_1		$p(a_0=0)$		$\Phi(a_0 \downarrow a_1 \to 1 \downarrow 1)$	
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	$0Sx_5$	$0Lx_3$	$0Rx_4$	$0Rx_4$	$0Lx_6$	$0Sx_7$	$1Rx_8$	$1Lx_9$
1	$1Sx_2$	$1Lx_2$	$1Lx_2$			$1Sx_2$		$1Sx_9$

	$p_Л$		A_2		$p_П$		A_3	
	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
0		$0Sx_{11}$	$1Rx_{11}$	$0Rx_{12}$		$0Sx_{15}$	$1Lx_{15}$	$0Lx_{16}$
1	$1Lx_{10}$	$1Sx_{17}$	$0Rx_{12}$	$1Sx_{13}$	$1Rx_{14}$	$1Sx_{21}$	$0Lx_{16}$	$1Sx_9$

	A_4				A_5			
	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}
0			$0Rx_{19}$	$0Lx_{20}$			$0Lx_{23}$	
1	$1Rx_{18}$	$0Rx_{19}$	$0Lx_{20}$	$1Sx_2$	$1Lx_{22}$	$0Lx_{23}$	$0Rx_{24}$	$0Rx_{24}$

можно изъять, так как в них мы попадаем непосредственно из $0Sx_5$, $0Sx_7$, $1Sx_9$, $1Sx_{13}$, $1Sx_{17}$, $1Sx_{21}$. После этого можно опустить состояния x_5, x_7, x_9, x_{13}, x_{17}, x_{21}. Мы получаем программу с 18 состояниями (см. табл. 12)*).

В дальнейшем, как правило, построение программ будет доводиться до II этапа — составления операторных схем, так как оставшаяся часть работы трудностей не вызывает.

Пользуясь принципом двойственности, легко построить программу, позволяющую находить правую единицу основного кода. Несколько усложняя идею, можно построить

*) В ней можно отождествить x_4 и x_{24}.

программы для нахождения правой или левой единицы i-го массива основного кода.

Определение. *Решеткой с шагом* l $(l \geqslant 2)$ называется последовательность ячеек ленты, номера которых сравнимы по модулю l. Всего имеется l решеток с шагом l.

Таблица 12

	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_6$	$\varkappa_8$	$\varkappa_{10}$	$\varkappa_{11}$	$\varkappa_{12}$
0	$0L\varkappa_6$	$0L\varkappa_3$	$0R\varkappa_4$	$0R\varkappa_4$	$1R\varkappa_8$	$1L\varkappa_8$	$0S\varkappa_{11}$	$1R\varkappa_{11}$	$0R\varkappa_{12}$
1	$1S\varkappa_2$	$1L\varkappa_2$	$1L\varkappa_2$		$1S\varkappa_2$	$1L\varkappa_{10}$	$1R\varkappa_{13}$	$0R\varkappa_{12}$	$1R\varkappa_{14}$
	$\varkappa_{14}$	$\varkappa_{15}$	$\varkappa_{16}$	$\varkappa_{18}$	$\varkappa_{19}$	$\varkappa_{20}$	$\varkappa_{22}$	$\varkappa_{23}$	$\varkappa_{24}$
0	$0S\varkappa_{15}$	$1L\varkappa_{15}$	$0L\varkappa_{16}$		$0R\varkappa_{19}$	$0L\varkappa_{20}$		$0L\varkappa_{23}$	$0R\varkappa_{24}$
1	$1L\varkappa_{22}$	$0L\varkappa_{16}$	$1L\varkappa_{10}$	$0R\varkappa_{19}$	$0L\varkappa_{20}$	$1S\varkappa_2$	$0L\varkappa_{23}$	$0R\varkappa_{24}$	

Лемма 1 (о моделировании на решетке). *Пусть* $\mathfrak{M}$ *— произвольная машина Тьюринга с программой* T *и* l *— произвольное целое число* $(l \geqslant 2)$. *Тогда можно построить машину* $\widetilde{\mathfrak{M}}$, *которая на решетке с шагом* l *работает так же, как исходная машина* $\mathfrak{M}$ *на всей ленте.*

Доказательство. По таблице T (см. табл. 13) строим таблицу $\widetilde{T}$, в которой к каждому состоянию $\varkappa_j$ добавлены вспомогательные состояния $\varkappa_j^1, \ldots, \varkappa_j^{l-1}$, $\varkappa_j^l, \ldots, \varkappa_j^{2l-2}$, предназначенные для прохождения ячеек вне решетки и запоминания характера движения (см. табл. 14), где

$$\varkappa_j^D = \begin{cases} \varkappa_j & \text{при} \quad D = S, \\ \varkappa_j^1 & \text{при} \quad D = R, \\ \varkappa_j^l & \text{при} \quad D = L. \end{cases}$$

Из данной таблицы видно, что машина $\widetilde{\mathfrak{M}}$, находясь в ячейке решетки и обозревая символ a в состоянии $\varkappa_i$, заменяет, как и машина $\mathfrak{M}$, символ a на c, совершает то же движение D и переходит в состояние $\varkappa_j^D$. Состояние $\varkappa_j^D$ зависит от характера движения: это будет $\varkappa_j$ при $D = S$, $\varkappa_j^1$ при $D = R$ и $\varkappa_j^l$ при $D = L$. В последних

двух случаях головка машины сходит с решетки и далее при движении вправо проходит состояния $\varkappa_j^1, \varkappa_j^2, \ldots, \varkappa_j^{l-1}$, а при движении влево — состояния $\varkappa_j^l, \varkappa_j^{l+1}, \ldots, \varkappa_j^{2l-2}$. Затем головка попадает на решетку в состоянии $\varkappa_j$, сместившись по решетке на одну ячейку. Таким образом, в случае движения R и L машина $\widetilde{\mathfrak{M}}$ делает на решетке

Таблица 13

	$\varkappa_1$	...	$\varkappa_i$	...	$\varkappa_j$	...	$\varkappa_r$
0							
...							
a			$cD\varkappa_j$				
...							
$k-1$							

то же самое, что и машина $\mathfrak{M}$ на всей ленте, но за l шагов.

Следствия. 1) *Если* $C_j(a) \equiv a$ *при* $a = 0, \ldots, k-1$ *и любом* j, *то машина* $\widetilde{\mathfrak{M}}$ *вне решетки не меняет записи ленты.*

2) *Если* $C_j(a) \equiv 0$ *при* $a = 0, \ldots, k-1$ *и любом* j, *то машина* $\widetilde{\mathfrak{M}}$ *в пределах рабочей зоны производит очистку ленты.*

3) *Если* $C_j(a) \equiv 1$ *при* $a = 0, \ldots, k-1$ *и любом* j, *то машина* $\widetilde{\mathfrak{M}}$ *в пределах рабочей зоны ставит* 1, *тем самым отмечает те ячейки вне решетки, в которых побывала головка.*

4) *Возможны смешанные ситуации, например,* $C_1(a) = C_{2l-2}(a) = 1$ *при* $a = 0, \ldots, k-1$ *и* $C_j(a) = a$ *в остальных случаях. В этом случае машина* $\widetilde{\mathfrak{M}}$ *ставит* 1 *в пределах рабочей зоны на соседней решетке, являющейся сдвигом исходной решетки на единицу вправо, и не меняет содержимого ленты вне этих двух решеток.*

Таблица 14

	...	x_i	...	x_j^1	x_j^2	...	x_j^{l-1}	x_j^l	...	x_j^{2l-2}	...
0	...	...	...	$C_1(0)\, Rx_j^2$	$C_2(0)\, Rx_j^3$	...	$C_{l-1}(0)\, Rx_j$	$C_l(0)\, Lx_j^{l+1}$	...	$C_{2l-2}(0)\, Lx_j$	...
...	...	...	...	...	...	...	...	...	...	...	...
a	...	cDx_j^D	...	$C_1(a)\, Rx_j^2$	$C_2(a)\, Rx_j^3$	...	$C_{l-1}(a)\, Rx_j$	$C_l(a)\, Lx_j^{l+1}$	...	$C_{2l-2}(a)\, Lx_j$	...
...	...	...	...	...	...	...	...	...	...	...	...
$k-1$	...	...	...	$C_1(k-1)\, Rx_j^2$	$C_2(k-1)\, Rx_j^3$	...	$C_{l-1}(k-1)\, Rx_j$	$C_l(k-1)\, Lx_j^{l+1}$	...	$C_{2l-2}(k-1)\, Lx_j$	...

Введем следующее обозначение. Если A есть некоторое преобразование на ленте, то $\bar{A}$ будет обозначать оператор, моделирующий A на некоторой решетке (его поведение вне решетки будет специально уточняться).

Перейдем теперь к описанию вспомогательных кодов. Мы различаем три вида *вспомогательных кодов*.

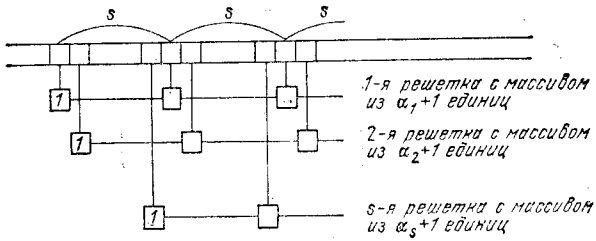


Рис. 6

а) *l-кратный код* определяется для произвольного набора $\alpha_1, \alpha_2, \ldots, \alpha_s$ чисел из расширенного натурального ряда следующим образом:

$$\ldots 0\underbrace{1\ldots1}_{l(\alpha_1+1)}U\underbrace{1\ldots1}_{l(\alpha_2+1)}U\ldots U\underbrace{1\ldots1}_{l(\alpha_s+1)}0\ldots,$$

где U — буферное слово длины l и $U=0U'$, т. е. U начинается с нуля;

б) *решетчатый код* определяется для произвольного набора $\alpha_1, \alpha_2, \ldots, \alpha_s$ чисел из расширенного натурального ряда. Это запись на ленте, которую можно разложить с помощью s решеток (последовательностей ячеек ленты), имеющих период s, на массивы из единиц, а именно: на первой решетке расположен массив из α_1+1 единиц, на второй решетке расположен массив из α_2+1 единиц и т. д., на s-й решетке расположен массив из α_s+1 единиц и начала этих массивов согласованы, т. е. идут на ленте подряд в соответствии с номерами решеток (см. рис. 6; на данном рисунке для наглядности каждая решетка изображена отдельно);

в) *квазиосновной код* определяется для произвольного основного кода $b_1b_2\ldots b_\nu$, где $b_1=b_\nu=1$, в виде следующей записи:

$$b_1U_1b_2U_2\ldots U_{\nu-1}b_\nu,$$

где

$$|U_1|=\ldots=|U_{\nu-1}|=l-1 \quad (l\geqslant 2).$$

Таким образом, в квазиосновном коде на решетке с шагом l расположен основной код, внутренние промежутки заполнены словами U_1, U_2, ..., $U_{\nu-1}$, а остальная часть ленты — пустая.

С машинными кодами и их преобразованиями связан ряд лемм.

Лемма 2 (о преобразовании основного кода в l-кратный код). *Пусть l — натуральное число ($l \geqslant 2$). Тогда можно построить машину Тьюринга, преобразующую основной код в соответствующий l-кратный с некоторым заданным буферным словом U, где $|U| = l$ и $U = 0U'$.*

Доказательство. I этап. Искомая машина будет постепенно на ленте правее основного кода строить l-кратный код. Для этого просматриваются слева направо символы основного кода и каждый символ основного кода на некотором шаге заменяется нулем, а справа пристраивается либо массив из l единиц, если в основном коде была 1, либо слово U, если в основном коде был 0. В процессе этого построения на ленте будет появляться слово, в котором между двумя соседними единицами не может стоять более l нулей. Для таких слов можно (как для основных кодов) построить машину, находящую его левый (соответственно правый) конец. Обозначим соответствующие перемещения головки на ленте через L и R.

Таким образом, работу машины можно описать более точно:

1) выходим на правый конец основного кода (оператор R);

2) отступая две клетки вправо, выписываем массив из l единиц. Данное преобразование обозначим через $\Phi(a^{\downarrow} \to \to a\,0\,0\underbrace{1\ldots1}_{l}{}^{\downarrow})$;

3) возвращаемся на левый конец основного кода (оператор L);

4) просматривая первые три символа $a'a''a'''$ с левого конца основного кода (а в последующем — в оставшейся части основного кода), вычисляем трехзначный предикат $p(a', a'', a''')$.

Если $a'a''a''' = 11a'''$, то имеем I режим и переходим к преобразованию, которое указывается стрелкой $\uparrow$ ($p^{\uparrow}$).

Если $a'a''a''' = 101$, то имеем II режим и переходим к оператору, следующему за p.

Если $a'a''a''' = 100$, то имеем III режим и переходим к преобразованию, которое указывается стрелкой $\downarrow$ ($p_{\downarrow}$).

I режим.

5) Стираем символ a' (оператор $0(a')$);

6) Выходим на правый конец слова (оператор R);

7) Пристраиваем непосредственно справа от слова массив из l единиц (оператор $\Phi(a^{\downarrow} \to a\underbrace{1\ldots1}_{l}{}^{\downarrow})$);

8) Возвращаемся на левый конец слова (оператор L) и переходим к 4).

II режим (первый символ a' — последний в массиве единиц основного кода, но имеется по крайней мере один не обработанный массив из единиц в основном коде).

9) Стираем символ $\cdot a'$ $(0(a'))$;

10) Выходим на правый конец слова (оператор R);

11) Пристраиваем непосредственно справа от слова буферное слово U и еще l единиц (оператор $\Phi(a^{\downarrow} \to aU\underbrace{1\ldots1}_{l}{}^{\downarrow})$);

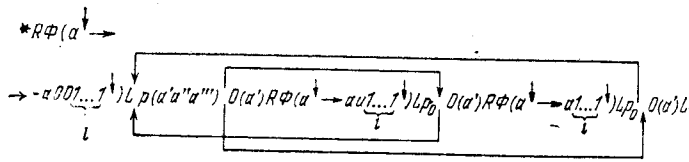
12) Возвращаемся на левый конец слова (оператор L) и переходим к 4).

III режим (первый символ a' является последним в основном коде).

13) Стираем символ a' $(0(a'))$;

14) Двигаясь вправо, выходим на левый конец построенного l-кратного кода и останавливаемся.

II этап. Мы получаем следующую операторную схему:



Реализация операторов этой схемы не вызывает затруднений и по схеме легко может быть построена программа машины. Лемма доказана.

Лемма 3 (о преобразовании решетчатого кода в основной). *Пусть s — натуральное число, $s \geqslant 2$. Тогда можно построить машину Тьюринга, которая преобразует произвольный решетчатый код с параметром s в соответствующий основной код.*

Доказательство. I этап. Сначала опишем идею работы машины. Обозревая в начальный момент левую единицу решетчатого кода, машина отступает далее влево на определенное расстояние и постепенно справа налево формирует там основной код путем «перетаскивания» кодов с решеток, начиная с s-й и кончая 1-й решеткой (см. рис. 7).

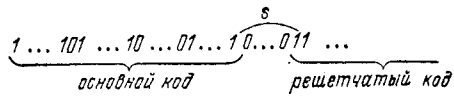


Рис. 7

Более точно преобразование можно характеризовать так: разобьем его на последовательность более простых преобразований

$$* A_0 A_s \dots A_1 \omega.$$

Преобразование A_0 (предварительная подготовка ленты). 1) Отступаем от начальной ячейки (левая единица решетчатого кода) влево на s ячеек (оператор L_1^s)*).

2) Непосредственно слева за этим промежутком вписываем единицу (оператор $\Phi(a^i \to 1^i a)$). Преобразование закончено.

Преобразование A_s («перетаскивание» массива единиц с s-й решетки). 1) От ячейки, в которой оператор A_0 поставил 1, смещаемся вправо на $2s$ ячеек (оператор R_1^{2s}). Мы попадаем на левый конец массива, расположенного на s-й решетке. Пусть a', a'' — первые два символа на s-й решетке (a' — левый конец).

2) Выясняем, является ли a' одновременно и правым концом кода путем вычисления предиката $\tilde{p}(a', a'')$ (с возвращением к a'):

$$\tilde{p}(a', a'') = \begin{cases} 0 \text{ при } a'' = 0 \ (a' \text{ — правый конец массива}), \\ 1 \text{ при } a'' = 1 \ (a' \text{ — не есть правый конец массива}). \end{cases}$$

*) В решетчатом коде подряд может стоять $s - 1$ нулей.

Если $\tilde{p}=1$ (a' — не правый конец массива на s-й решетке), то:

3) Движемся до правого конца массива на этой решетке (оператор $\bar{R}$).

4) Стираем последний символ a в массиве (оператор $0(a)$).

5) Возвращаемся на левый конец массива (оператор $\bar{L}$).

6) Смещаемся влево еще на $2s$ ячеек (оператор L_1^{2s}) — мы попадаем на правый конец основного кода (вернее — построенного куска основного кода).

7) Выходим на левый конец основного кода (оператор L).

8) Приписываем слева к основному коду символ 1 (оператор $\Phi(a^{\downarrow}\to 1^{\downarrow}a)$).

9) Возвращаемся на правый конец основного кода (оператор R), после чего переходим к 1).

Если $\tilde{p}=0$ (a' — правый конец массива на s-й решетке), то:

10) Стираем символ a' (оператор $0(a')$).

11) Смещаемся влево на $2s$ ячеек (оператор L_1^{2s}), мы попадаем на правый конец основного кода.

12) Выходим на левый конец основного кода (оператор L).

13) Приписываем слева к основному коду символы 10 (оператор $\Phi(a^{\downarrow}\to 1^{\downarrow}0a)$).

14) Возвращаемся на правый конец основного кода (R).

Преобразование закончено.

Преобразование A_i $(1<i\leqslant s)$ («перетаскивание» массива единиц с i-й решетки). Выполняется так же, как и A_s с заменой операторов R_1^{2s} и L_1^{2s} соответственно на R_1^{s+i} и L_1^{s+i}, что обеспечивает попадание на i-ю решетку.

Преобразование A_1 («перетаскивание» массива единиц с 1-й решетки). Выполняется так же, как и A_s с заменой операторов R_1^{2s} и L_1^{2s} соответственно на R_1^{s+1} и L_1^{s+1} и изъятием операторов $\Phi(a^{\downarrow}\to 1^{\downarrow}0a)$ и R (см. п.п. 13) и 14)), поскольку эти операторы осуществляют подготовку к следующему преобразованию, а A_1 является последним.

II этап. Очевидно, мы имеем следующие оператор-

ные схемы для преобразований A_0, A_s, ..., A_i, ..., A_1:

$$A_0 = *L_1^s \Phi(a^{\downarrow} \to 1^{\downarrow} a)\omega,$$

$$A_s = *\boxed{R_1^{2s}\tilde{p}(a'a'')\boxed{\tilde{R}0(a)\tilde{L}L_1^{2s}L\Phi(a^{\downarrow}\to 1^{\downarrow}a)Rp_0}\uparrow 0(a')L_1^{2s}L\Phi(a^{\downarrow}\to 1^{\downarrow}0a)R\omega},$$

. . . . . .

$$A_i = *\boxed{R_1^{s+i}\tilde{p}(a'a'')\boxed{\tilde{R}0(a)\tilde{L}L_1^{s+i}L\Phi(a^{\downarrow}\to 1^{\downarrow}a)Rp_0}\uparrow 0(a')L_1^{s+i}L\Phi(a^{\downarrow}\to 1^{\downarrow}0a)R\omega},$$

. . . . . .

$$A_1 = *\boxed{R_1^{s+1}\tilde{p}(a'a'')\boxed{\tilde{R}0(a)\tilde{L}L_1^{s+1}L\Phi(a^{\downarrow}\to 1^{\downarrow}a)Rp_0}\uparrow 0(a')L_1^{s+1}L\omega}.$$

Лемма доказана.

Лемма 4 (о преобразовании квазиосновного кода в основной). *Для любого натурального числа* l $(l \geqslant 2)$ *можно построить машину Тьюринга, которая произвольный квазиосновной код* $b_1U_1b_2 \ldots U_{v-1}b_v$, *где* $|U_1| = \ldots = |U_{v-1}| = l-1$, *преобразует в соответствующий основной код* $b_1b_2 \ldots b_v$.

Доказательство. I этап. Искомая машина сначала отступает на некоторый промежуток вправо от квазиосновного кода и затем постепенно там формирует основной код путем «перетаскивания» основного кода с решетки.

Более точно преобразование состоит в следующем.

1) В начальный момент обозревается символ b_1, т. е. символ в узле решетки с шагом l, на которой расположен основной код. Движемся по решетке на правый конец основного кода (оператор $\tilde{R}$), осуществляя стирание буферных слов U_1, ..., U_{v-1}, лежащих вне решетки (см. следствие 2 леммы 1).

2) Справа от правого конца (символ b_v) вписываем $3l-1$ нулей*) и одну единицу (оператор $\Phi(b_v^{\downarrow} \to b_v \underbrace{0 \ldots 0}_{3l-1} 1^{\downarrow})$), после чего попадаем на ту же решетку.

3) Отыскиваем левый конец основного кода (оператор $\tilde{L}$).

---

*) В квазиосновном коде может подряд стоять $2l-1$ нулей.

4) Пусть $a'a''a'''$ — первые три символа в основном коде (в начальный момент это $b_1b_2b_3$). Вычисляем предикат $\tilde{p}(a'a''a''')$, принимающий три значения. При $a''=1$ (основной режим) переходим к выполнению следующего за $\tilde{p}$ оператора. При $a''=0$, $a'''=1$ a' является последней единицей в массиве и так как $a'''=1$, то имеется по крайней мере еще один массив — переходим к выполнению оператора, указываемого стрелкой с пометкой $a''=0$, $a'''=1$. При $a''=a'''=0$ a' является последней единицей основного кода (т. е. $a'=b_\nu$) — переходим к выполнению оператора, указываемого стрелкой с пометкой $a''=a'''=0$.

Основной режим $a''=1$.

5) Стираем символ a' (оператор $0(a')$).

6) Движемся на правый конец основного кода (оператор $\tilde{R}$).

7) Перемещаемся вправо еще на $3l$ ячеек (оператор R_1^{3l}).

8) Выходим на правый конец формируемого кода (оператор R).

9) Приписываем справа единицу (оператор $\Phi(a^{\downarrow}\to a1^{\downarrow})$).

10) Движемся на левый конец формируемого кода (оператор L) и переходим к 3).

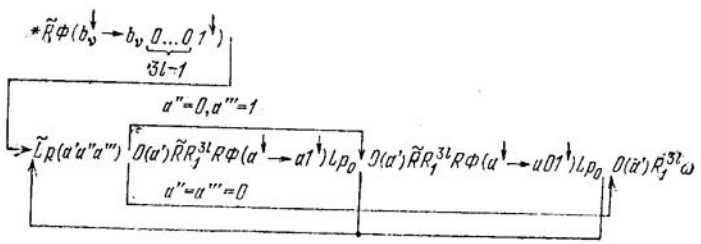
Режим $a''=0$, $a'''=1$.

Выполняем то же, что и в предыдущем режиме 5)—10), кроме пункта 9), где выполняем оператор $\Phi(a^{\downarrow}\to a01^{\downarrow})$.

Режим $a''=a'''=0$.

Здесь стираем символ a' (оператор $0(a')$) и, смещаясь вправо на $3l$ ячеек (оператор R_1^{3l}), попадаем на левый конец искомого основного кода и останавливаемся.

II этап. Мы имеем следующую операторную схему:



§ 4. Вычислимые функции

Выше мы ввели систему $P_{\aleph_0}$, содержащую все константы из $E_{\aleph_0}$ и все функции, определенные на наборах чисел из расширенного натурального ряда $E_{\aleph_0}$ и принимающие значения на $E_{\aleph_0}$. Сейчас мы определим более широкую, чем $P_{\aleph_0}$, систему функций.

Пусть $f(x_1, \ldots, x_n)$ — функция, определенная на подмножестве E_f множества всех наборов $(\alpha_1, \ldots, \alpha_n)$ чисел из расширенного натурального ряда $E_{\aleph_0}$ и принимающая значения также из $E_{\aleph_0}$ (вне множества E_f функция считается неопределенной). Такого рода функции будем называть *частичными функциями счетнозначной логики.* Обозначим через $P^{\text{ч}}_{\aleph_0}$ множество всех частичных функций счетнозначной логики.

Как и в случае не всюду определенных функций из P_2 (см. гл. 3), можно ввести понятие несущественной переменной.

Определение. Переменная x_i называется *несущественной* для функции $f(x_1, \ldots, x_n)$ из $P^{\text{ч}}_{\aleph_0}$, если существует функция f' из $P_{\aleph_0}$ такая, что

$$f'(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$$

на E_f и переменная x_i не существенна для $f'(x_1, \ldots, x_n)$.

В дальнейшем частичные функции будем рассматривать с точностью до несущественных переменных, относительно которых множество E_f цилиндрично. В этом случае существует доопределение функции f, т. е. функция f' из $P_{\aleph_0}$, которая несущественно зависит от всех таких переменных.

Определение. Функция $f(x_1, \ldots, x_n)$, где $f \in P^{\text{ч}}_{\aleph_0}$, называется *вычислимой*, если существует машина Тьюринга $\mathfrak{M}$ такая, что:

а) при $(\alpha_1, \ldots, \alpha_n) \in E_f$ машина $\mathfrak{M}$, будучи применена к основному коду для $(\alpha_1, \ldots, \alpha_n)$ и находясь в начальном состоянии над его левой единицей, останавливается и в заключительном состоянии на ленте выдает код для $f(\alpha_1, \ldots, \alpha_n)$;

б) при $(\alpha_1, \ldots, \alpha_n) \notin E_f$ машина $\mathfrak{M}$, будучи применена к основному коду для $(\alpha_1, \ldots, \alpha_n)$ и находясь в начальном состоянии над его левой единицей, либо не останав-

ливается, либо останавливается, но при этом запись на ленте отлична от кода любого числа из $E_{\aleph_0}$.

Замечание. Константы из $E_{\aleph_0}$ можно также считать вычислимыми функциями с пустым множеством переменных в следующем смысле.

Пусть $\gamma \in E_{\aleph_0}$. Рассмотрим машину, задаваемую табл. 15. Поскольку γ — константа, то в начальном положении лента считается пустой. Очевидно, машина, начи-

Таблица 15

	x_1	x_2	. . .	$x_{\gamma+1}$	$x_{\gamma+2}$
0	$1Rx_2$	$1Rx_3$	. . .	$1Sx_{\gamma+2}$	$0Rx_{\gamma+1}$
1					$1Lx_{\gamma+2}$

Таблица 16

	x_1	x_2
0	$1Sx_2$	
1	$0Rx_1$	

ная от исходной ячейки, движется вправо и формирует массив из $\gamma + 1$ единиц — код γ, и затем возвращается на левый конец массива.

Приведем пример вычислимой функции.

Пример 6. Покажем, что функция $0(x) \equiv 0$ вычислима. Для этого возьмем машину, определяемую табл. 16. Очевидно, что эта машина «реализует» функцию $0(x) \equiv 0$. Заметим, что данная машина «реализует» также функцию $f(x_1, x_2) = x_2 + 1$ и константу 0 (как функцию, зависящую от пустого множества переменных).

Обозначим через $P_{\text{выч}}$ класс всех вычислимых функций. Очевидно, $P_{\text{выч}} \subseteq P^{\text{ч}}_{\aleph_0}$.

Определение. Машина Тьюринга $\mathfrak{M}$ реализует (вычисляет) функцию $f(x_1, \ldots, x_n)$ (из класса $P_{\text{выч}}$) *правильным образом*, если:

а) при $(\alpha_1, \ldots, \alpha_n) \in E_f$ машина $\mathfrak{M}$, будучи применена к основному коду для $(\alpha_1, \ldots, \alpha_n)$ и находясь в начальном состоянии над его левой единицей, останавливается и в заключительном состоянии на ленте выдает код для $f(\alpha_1, \ldots, \alpha_n)$; при этом останов происходит над левой единицей кода для $f(\alpha_1, \ldots, \alpha_n)$;

б) при $(\alpha_1, \ldots, \alpha_n) \notin E_f$ машина $\mathfrak{M}$, будучи применена к основному коду для $(\alpha_1, \ldots, \alpha_n)$ и находясь в начальном состоянии над его левой единицей, не останавливается.

Легко видеть, что машина (см. табл. 16) реализует $0(x) \equiv 0$ правильным образом.

Лемма 5. *Если $f(x_1, \ldots, x_n)$ — вычислимая функция, то существует машина Тьюринга, которая вычисляет ее правильным образом.*

Доказательство. Пусть $\mathfrak{M}'$ — машина, вычисляющая функцию $f(x_1, \ldots, x_n)$. Соответствующее преобразование записи ленты, положения головки и состояний обозначим через A'. Рассмотрим преобразование

$$A = *K_1 \widetilde{A'} \overline{\overline{p}} 0_2 K_3 \omega.$$

Здесь K_1 — преобразование кода для $(\alpha_1, \ldots, \alpha_n)$ в удвоенный код с буферным словом 01. На первой решетке будет код для $(\alpha_1, \ldots, \alpha_n)$, на второй — сплошной массив из единиц.

$\widetilde{A'}$ — преобразование, моделирующее на первой решетке преобразование A'; вне этой решетки $\widetilde{A'}$ в рабочей зоне ставит символ 1.

p — предикат, выясняющий вид слова на первой решетке после работы оператора $\widetilde{A'}$. Просмотр слова осуществляется при помощи второй решетки, которая своим массивом из единиц отмечает зону обследования на первой решетке. Полагаем $p = 1$, если слово является массивом из единиц, и $p = 0$, если в слове найдутся две единицы, между которыми имеется нуль, или в нем нет единиц вообще.

0_2 — преобразование, которое стирает все единицы на второй решетке и останавливается над левой единицей слова, расположенного на первой решетке.

K_3 — преобразование квазиосновного кода в основной.

Из данной схемы видно, что в случае, когда после осуществления преобразования $\widetilde{A'}$ запись на первой решетке будет отлична от массива из единиц, преобразование зацикливается, так как будет все время вычисляться предикат p.

Таблица 17а

s	x_1	x_2
0	$1Sx_2$	
1	$1Lx_1$	

Машина $\mathfrak{M}$, соответствующая преобразованию A, и будет искомой. Лемма доказана.

В дальнейшем для вычислимых функций будем использовать исключительно машины, вычисляющие их правильным образом.

Теперь перейдем к описанию некоторых простейших вычислимых функций. Рассмотрим следующие функции:

1) константа 0; 2) $S(x)=x+1$; 3) $I_m^n(x_1, \ldots, x_n)=x_m$, где $1 \leqslant m \leqslant n$.

Покажем, что данные функции вычислимы. Это уже сделано для константы 0. Вычислимость функций $S(x)$ и I_m^n следует из того, что они реализуемы следующими машинами (см. табл. 17а, б). Машина для $I_m^n(x_1, \ldots, x_n)$ идет направо (в начальном состоянии обозревается, как

Таблица 17б

I_m^n	x_1	$\ldots$	x_{m-1}	x_m
0	$0Rx_2$	$\ldots$	$0Rx_m$	$0Rx_{m+1}$
1	$0Rx_1$	$\ldots$	$0Rx_{m-1}$	$1Rx_m$

I_m^n	x_{m+1}	$\ldots$	x_n	x_{n+1}	x_{n+2}	x_{n+3}
0	$0Rx_{m+2}$	$\ldots$	$0Lx_{n+1}$	$0Lx_{n+1}$	$0Rx_{n+3}$	
1	$0Rx_{m+1}$	$\ldots$	$0Rx_n$	$1Lx_{n+2}$	$1Lx_{n+2}$	

всегда, левая единица основного кода) и стирает все массивы основного кода для $(\alpha_1, \ldots, \alpha_n)$, кроме m-го, затем возвращается влево и встает над левой единицей оставшегося массива.

§ 5. Операции C, Пр и μ

На множестве $P^{\text{ч}}_{\text{к}_0}$ определим три операции: C (суперпозиция), Пр (примитивная рекурсия) и μ (минимизация).

Операция суперпозиции вводится так же, как и для предыдущих функциональных систем: сначала определяется понятие формулы $\mathfrak{A}(x_1, \ldots, x_n)$ над данной системой функций из $P^{\text{ч}}_{\text{к}_0}$, потом каждой формуле $\mathfrak{A}$ сопоставляется функция $f_{\mathfrak{A}}(x_1, \ldots, x_n)$, принадлежащая $P^{\text{ч}}_{\text{к}_0}$. При этом, если на наборе $(\alpha_1, \ldots, \alpha_n)$ окажется, что одна из функций, входящая в $\mathfrak{A}$, будет неопределенной, то считаем, что $f_{\mathfrak{A}}(\alpha_1, \ldots, \alpha_n)$ будет также неопределенной. Более точно, пусть

$$\Phi(x_1, \ldots, x_n)=f(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Возьмем произвольный набор $(\alpha_1, \ldots, \alpha_n)$ чисел из расширенного натурального ряда. Если на этом наборе определены функции $f_1, \ldots, f_m$ и функция f определена на наборе $(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$, то Φ определена на $(\alpha_1, \ldots, \alpha_n)$ и $\Phi(\alpha_1, \ldots, \alpha_n) = f(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$; в противном случае Φ не определена на наборе $(\alpha_1, \ldots, \alpha_n)$.

Операция *примитивной рекурсии* определяется следующим образом.

Пусть $\varphi(x_1, \ldots, x_n)$ и $\psi(x_1, \ldots, x_n, x_{n+1}, x_{n+2})$*) — произвольные функции из $P^{\text{ч}}_{\aleph_0}$. Построим функцию $f(x_1, \ldots, x_n, x_{n+1})$, используя «схему» примитивной рекурсии:

$$f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n),$$

$$f(x_1, \ldots, x_n, y+1) = \psi(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)).$$

Пусть $(\alpha_1, \ldots, \alpha_{n+1})$ — произвольный набор чисел из $E_{\aleph_0}$. Полагаем

$$f(\alpha_1, \ldots, \alpha_n, 0) = \varphi(\alpha_1, \ldots, \alpha_n).$$

Если φ на этом наборе не определена, то считаем, что не определена $f(\alpha_1, \ldots, \alpha_n, 0)$, а также $f(\alpha_1, \ldots, \alpha_n, y)$ при любом y. В противном случае полагаем

$$f(\alpha_1, \ldots, \alpha_n, 1) = \psi(\alpha_1, \ldots, \alpha_n, 0, f(\alpha_1, \ldots, \alpha_n, 0)).$$

Если правая часть не определена, то считаем, что $f(\alpha_1, \ldots, \alpha_n, 1)$, а также $f(\alpha_1, \ldots, \alpha_n, y)$ не определены при любом y, $y \geqslant 1$ и т. д.

Через конечное число шагов мы либо определим $f(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$, либо установим, что на этом наборе f не определена.

Из данного рассуждения видно, что если $f(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$ не определена, то при $\beta \geqslant \alpha_{n+1}$ не определена будет также и $f(\alpha_1, \ldots, \alpha_n, \beta)$. Про функцию f будем говорить, что она получена из функций φ и ψ при помощи операции примитивной рекурсии.

Пример 7. Покажем, что функция $f(x_1, x_2) = x_1 + x_2$ может быть получена через примитивную рекурсию из простейших вычислимых функций.

---

*) Некоторые из переменных у φ и ψ могут отсутствовать.

В самом деле,

$$f(x_1, 0) = I'_1(x_1),$$
$$f(x_1, y+1) = S(f(x_1, y)).$$

Замечание. Операция Пр позволяет для каждой функции $\varphi(x_1, \ldots, x_n)$ вводить несущественные переменные, а именно:

$$f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = \varphi(x_1, \ldots, x_n).$$

Операция *минимизации* определяется следующим образом. Пусть $\varphi(x_1, \ldots, x_{n-1}, x_n)$ — произвольная функция из $P^{\text{ч}}_{\text{к}_0}$. Построим функцию $f(x_1, \ldots, x_{n-1}, x_n)$ через оператор минимизации

$$f(x_1, \ldots, x_n) = \mu_y(\varphi(x_1, \ldots, x_{n-1}, y) = x_n),$$

что означает, что для произвольного набора $(\alpha_1, \ldots, \alpha_n)$ составляется уравнение

$$\varphi(\alpha_1, \ldots, \alpha_{n-1}, y) = \alpha_n.$$

а) Если существует y из $E_{\text{к}_0}$, являющееся решением этого уравнения, то берем минимальное из решений и обозначим его через μ_y. Если значения

$$\varphi(\alpha_1, \ldots, \alpha_{n-1}, 0), \ldots, \varphi(\alpha_1, \ldots, \alpha_{n-1}, \mu_y - 1)$$

также определены, то полагаем

$$f(\alpha_1, \ldots, \alpha_{n-1}, \alpha_n) = \mu_y.$$

б) В противном случае, т. е. в случае, когда либо уравнение не имеет решений, либо хотя бы одно из значений $\varphi(\alpha_1, \ldots, \alpha_{n-1}, 0), \ldots, \varphi(\alpha_1, \ldots, \alpha_{n-1}, \mu_y - 1)$ не определено, функция $f(\alpha_1, \ldots, \alpha_n)$ также не определена.

Про функцию f говорят, что она получена из функции φ при помощи операции минимизации.

Пример 8. Пусть $\varphi(x) = x + 1$. Определим через операцию μ функцию $f(x)$:

$$f(x) = \mu_y(\varphi(y) = x).$$

Ясно, что

$$f(x) = \begin{cases} \text{не определена} & \text{при } x = 0, \\ x - 1 & \text{при } x > 0. \end{cases}$$

Данные операции позволяют построить три следующие функциональные системы.

I. Множество $P_{чр}$ всех функций, которые можно получить из системы функций $\{0, S(x), I_m^n(x_1, \ldots, x_n), 1 \leqslant m \leqslant n, n = 1, 2, \ldots\}$ при помощи операций C, Пр и μ, называемое *классом частично-рекурсивных функций.*

II. *Класс рекурсивных функций*, т. е. множество $P_р$ всех всюду определенных функций из $P_{чр}$.

III. *Класс примитивно-рекурсивных функций*, т. е. множество $P_{пр}$ всех функций, которые можно получить из системы $\{0, S(x), I_m^n(x_1, \ldots, x_n), 1 \leqslant m \leqslant n, n = 1, 2, \ldots\}$ при помощи операций C и Пр.

Очевидно, что

$$P_{пр} \subseteq P_р \subseteq P_{чр} \subseteq P^{\text{ч}}_{\aleph_0}.$$

Предыдущий пример показывает, что класс $P_{чр}$ существенно шире, чем класс $P_р$. Можно показать, что и класс $P_р$ существенно шире, чем класс $P_{пр}$.

Рассмотрим примеры примитивно-рекурсивных функций. Возьмем функции

$$\mathrm{Sg}(x),\ \overline{\mathrm{Sg}}(x),\ [x/2],\ 2^x,\ x_1 \dot{-} x_2 \text{ и } x_1 \cdot x_2,$$

где

$$\mathrm{Sg}(x) = \begin{cases} 0 \text{ при } x = 0, \\ 1 \text{ при } x \neq 0, \end{cases} \quad \overline{\mathrm{Sg}}(x) = \begin{cases} 1 \text{ при } x = 0, \\ 0 \text{ при } x \neq 0, \end{cases}$$

$$x_1 \dot{-} x_2 = \begin{cases} 0 & \text{при } x_1 < x_2, \\ x_1 - x_2 & \text{при } x_1 \geqslant x_2. \end{cases}$$

Функции $[x/2]$, 2^x и $x_1 \cdot x_2$ имеют обычный смысл: целая часть $x/2$, показательная функция и умножение. Их примитивная рекурсивность вытекает из следующих соотношений:

$$\begin{cases} \mathrm{Sg}(0) = 0, \\ \mathrm{Sg}(x+1) = 1, \end{cases} \qquad \begin{cases} \overline{\mathrm{Sg}}(0) = 1, \\ \overline{\mathrm{Sg}}(x+1) = 0, \end{cases}$$

$$\begin{cases} x_1 \cdot 0 = 0, \\ x_1(x_2+1) = x_1 x_2 + x_1, \end{cases} \qquad \begin{cases} 2^0 = 1, \\ 2^{x+1} = 2 \cdot 2^x. \end{cases}$$

Здесь константа 1 получается суперпозицией 0 и $S(x)$, $x_1 + x_2$ примитивно-рекурсивна, а $2x$ получается из нее

суперпозицией;

$$\begin{cases} 0 \dot{-} 1 = 0, \\ (x+1) \dot{-} 1 = x, \end{cases} \quad \begin{cases} x_1 \dot{-} 0 = x_1, \\ x_1 \dot{-} (x_2 + 1) = (x_1 \dot{-} x_2) \dot{-} 1, \end{cases}$$

$$\begin{cases} \left[\frac{0}{2}\right] = 0, \\ \left[\frac{x+1}{2}\right] = x \dot{-} \left[\frac{x}{2}\right]. \end{cases}$$

Здесь $x \dot{-} 1$ — вспомогательная функция.

Данные примитивно-рекурсивные функции позволяют строить многие другие примитивно-рекурсивные функции. Например, $f(x_1, \ldots, x_n)$, равная нулю за исключением конечного числа точек, в которых ее значения принадлежат $E_{\aleph_0}$, — примитивно-рекурсивна.

В самом деле, пусть

$$f(x_1, \ldots, x_n) = \begin{cases} \beta^i \text{ при } (x_1, \ldots, x_n) = (\alpha^i_1, \ldots, \alpha^i_n) & (i = 1, \ldots, s) \\ 0 \text{ в остальных точках} \end{cases}$$

и $\beta^i \in E_{\aleph_0} (i = 1, \ldots, s)$.

Рассмотрим функции $j_i(x)$, где

$$j_i(x) = \begin{cases} 1 \text{ при } x = i, \\ 0 \text{ при } x \neq i. \end{cases} \quad i = 0, 1, \ldots$$

Очевидно,

$$j_i(x) = \mathrm{Sg}(x \dot{-} (i-1)) \ \overline{\mathrm{Sg}}(x \dot{-} i) \text{ при } i \neq 0 \text{ и}$$

$$j_0(x) = \overline{\mathrm{Sg}}(x).$$

Положим

$$j_{i_1, \ldots, i_n}(x_1, \ldots, x_n) = \begin{cases} 1 \text{ при } (x_1, \ldots, x_n) = (i_1, \ldots, i_n), \\ 0 \text{ в остальных случаях.} \end{cases}$$

Мы имеем

$$j_{i_1, \ldots, i_n}(x_1, \ldots, x_n) = j_{i_1}(x_1) \ldots j_{i_n}(x_n),$$

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{s} \beta^i j_{\alpha^i_1 \ldots \alpha^i_n}(x_1, \ldots, x_n).$$

Последнее является аналогом разложения в дизъюнктивную нормальную форму.

В заключение заметим, что операции C и Пр, примененные к всюду определенным функциям, дают всюду определенные функции. Отсюда вытекает, что класс $P_р$ замкнут относительно операций C и Пр.

Далее мы займемся изучением связи классов $P_{выч}$ и $P_{чр}$. Основная цель состоит в установлении тождественности этих классов.

§ 6. Вычислимые функции и операции C, Пр, μ

Теперь мы займемся изучением особенностей операций C, Пр и μ над вычислимыми функциями.

Как и в случае предыдущих функциональных систем, мы будем рассматривать функции $f(x_1, \ldots, x_n)$ с точностью до добавлений и изъятий несущественных переменных и, более точно, несущественных переменных определенного вида. Это связано с тем, что вычислимая функция $f(x_1, \ldots, x_n)$ определена на некотором множестве E_f, которое не обязательно совпадает с множеством всех наборов $(\alpha_1, \ldots, \alpha_n)$ чисел из расширенного натурального ряда. В этом случае, если рассматривать несущественную переменную по аналогии с функциями из P_k как переменную, от которой для наборов из E_f функция не зависит, то возникают некоторые трудности. Например, процесс изъятия несущественных переменных может быть неоднозначным. (См. соответствующее рассуждение для не всюду определенных функций из P_k в гл. 3.) Ввиду этого переменную x_i функции $f(x_1, \ldots, x_n)$ из $P_{выч}$ мы будем называть *несущественной* (в узком смысле), если: 1) E_f цилиндрично по x_i; 2) f для наборов из E_f не зависит от x_i.

Операция удаления несущественной переменной (см. с. 12) уточняется следующим образом. Пусть для простоты $f(x_1, \ldots, x_n)$ имеет несущественную переменную x_n. По определению E_f цилиндрично по x_n и f для наборов из E_f не зависит от x_n. Рассмотрим функцию $g(x_1, \ldots, x_{n-1})$ с областью определения E_g такую, что:

1) E_g — проекция E_f на подпространство $(x_1, \ldots, x_{n-1})$; последнее в силу цилиндричности E_f по x_n эквивалентно условию $(\alpha_1, \ldots, \alpha_{n-1}) \in E_g$ тогда и только тогда, когда для любого $\alpha_n \in E_{\aleph_0}$ $(\alpha_1, \ldots, \alpha_{n-1}, \alpha_n) \in E_f$;

2) для любого $(\alpha_1, \ldots, \alpha_{n-1})$ из E_g

$$g(\alpha_1, \ldots, \alpha_{n-1}) = f(\alpha_1, \ldots, \alpha_{n-1}, 0).$$

Операция введения несущественной переменной выглядит так. Пусть $f(x_1, \ldots, x_n)$ — вычислимая функция с областью определения E_f. Рассмотрим функцию $h(x_1, \ldots, x_n, x_{n+1})$ с областью определения E_h такую, что:

1) E_h — цилиндр по x_{n+1} с основанием E_f, т. е. $(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \in E_h$ тогда и только тогда, когда $(\alpha_1, \ldots, \alpha_n) \in E_f$;

2) для любого $(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \in E_h$

$$h(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) = f(\alpha_1, \ldots, \alpha_n).$$

Лемма 6. *Из вычислимой функции при добавлении и изъятии несущественных переменных получается вычислимая функция.*

Доказательство. Справедливость леммы докажем для частных случаев.

а) Пусть $g(x_1, \ldots, x_{n-1})$ получена из $f(x_1, \ldots, x_{n-1}, x_n)$ путем изъятия несущественной переменной x_n. Рассмотрим преобразование

$$\text{код}\ (\alpha_1, \ldots, \alpha_{n-1}) \to \text{код}(\alpha_1, \ldots, \alpha_{n-1}, 0) \to$$
$$\to \text{код}\, f(\alpha_1, \ldots, \alpha_{n-1}, 0).$$

Соответствующая машина Тьюринга, очевидно, вычисляет функцию $g(x_1, \ldots, x_{n-1})$.

б) Пусть $h(x_1, \ldots, x_n, x_{n+1})$ получена из $f(x_1, \ldots, x_n)$ путем добавления несущественной переменной x_{n+1}. Рассмотрим преобразование

$$\text{код}(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \to \text{код}(\alpha_1, \ldots, \alpha_n) \to \text{код}\, f(\alpha_1, \ldots, \alpha_n).$$

Соответствующая машина Тьюринга вычисляет функцию $h(x_1, \ldots, x_n, x_{n+1})$.

Общий случай сводится к доказанным с использованием следствия приводимой ниже леммы.

Лемма 7*). *Если*

$$f(x_1, \ldots, x_m),\ f_1(x_1, \ldots, x_n),\ \ldots,\ f_m(x_1, \ldots, x_n)$$

---

*) В доказательстве леммы существенно, что вычислимая функция может быть реализована машиной, вычисляющей ее правильным образом.

вычислимы, то функция

$$f(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$$

также вычислима.

Доказательство. Рассмотрим преобразование

$$*K_1A_1A_2K_2A_3A_4\omega.$$

K_1 — преобразование кода $(\alpha_1, \ldots, \alpha_n)$ в $(m+1)$-кратный код с буферным словом $U = \underbrace{0 \ldots 01}_{m+1}$. Очевидно, что мы получаем на первых m решетках с шагом $m+1$ коды, совпадающие с кодом $(\alpha_1, \ldots, \alpha_n)$, а на $m+1$-й решетке — сплошной массив из единиц.

A_1 преобразует код $(\alpha_1, \ldots, \alpha_n)$

на 1-й решетке	в код	$f_1(\alpha_1, \ldots, \alpha_n)$,
на 2-й	»	$f_2(\alpha_1, \ldots, \alpha_n)$,
. . . . .		
на m-й	»	$f_m(\alpha_1, \ldots, \alpha_n)$

и на $m+1$-й решетке всюду, где побывает головка, ставится 1. Это преобразование выполняется путем использования m раз машин, моделирующих вычисление функций $f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, m)$ (см. следствие 4 к лемме 1).

A_2 осуществляет «выравнивание» кодов $f_i(\alpha_1, \ldots, \alpha_n)$ на решетках $i (i = 1, \ldots, m)$. Для этого на $m+1$-й решетке находят левую единицу и сдвигаются влево от нее на $3m+2$ ячеек *). Мы попадаем на первую решетку и осуществляем сдвиг кода $f_1(\alpha_1, \ldots, \alpha_n)$ к этой ячейке (моделирование машины, осуществляющей сдвиг влево). Затем из ячейки, в которой находится левая единица на 1-й решетке, смещаемся вправо на одну ячейку — мы попадаем на вторую решетку. Аналогичным образом производим сдвиг кода $f_2(\alpha_1, \ldots, \alpha_n)$ к этой ячейке и т. д. После сдвига кода $f_m(\alpha_1, \ldots, \alpha_n)$ на m-й решетке головка обозревает левую единицу на m-й решетке, и мы очищаем отрезок $m+1$-й решетки левее этой ячейки, а затем воз-

*) Левая единица на $m+1$-й решетке может быть правее левой единицы на 1-й решетке на m ячеек.

вращаемся к левой единице на 1-й решетке. Мы получили решетчатый код с параметром $s = m + 1$.

K_2 осуществляет преобразование решетчатого кода в основной.

A_3 в основном коде стирает последний $m+1$-й массив и возвращает головку к левой единице кода. Мы имеем на ленте код $(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$.

A_4 преобразует этот код в код $f(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$.

Очевидно, что данное преобразование выполнимо тогда и только тогда, когда значение $f(f_1, \ldots, f_m)$ определено. Здесь, по существу, используется вычислимость функций $f, f_1, \ldots, f_m$ машинами правильным образом.

Таким образом, машина, осуществляющая это преобразование, и будет искомой машиной. Лемма доказана.

Следствие 1. *Пусть* $\begin{pmatrix} 1 & 2 & \ldots & m \\ i_1 & i_2 & \ldots & i_m \end{pmatrix}$ *– произвольная подстановка. Тогда*

$$f(I^m_{i_1}(x_1, \ldots, x_m), I^m_{i_2}(x_1, \ldots, x_m), \ldots, I^m_{i_m}(x_1, \ldots, x_m)) = f(x_{i_1}, \ldots, x_{i_m}).$$

Это означает, что функция, получаемая из вычислимой функции путем перестановки переменных, вычислима.

Следствие 2. *Лемма 7 легко обобщается на случай, когда функции* $f_1, \ldots, f_m$ *зависят не от всех переменных* $x_1, \ldots, x_n$.

Последнее достигается путем добавления всех недостающих несущественных переменных (лемма 6) из функций $f_1, \ldots, f_m$ и применения леммы 7.

Теорема 1. *Класс* $P_{\text{выч}}$ *замкнут относительно операции суперпозиции.*

Доказательство основано на использовании леммы 7 и того, что тождественная функция $I^1_1(x_1)$ принадлежит классу $P_{\text{выч}}$ (см. замечание 3 на стр. 18).

При рассмотрении операций Пр и μ мы будем иметь дело с анализом и преобразованиями кодов, расположенных на решетках с некоторым шагом l. В связи с этим рассмотрим три оператора, которые содержательно можно охарактеризовать так:

оператор $p_>(\beta, \beta')$ производит сравнение двух чисел β и β' $(\beta \geqslant \beta')$, расположенных на двух решетках:

$$p_>(\beta, \beta') = \begin{cases} 1 \text{ при } \beta > \beta', \\ 0 \text{ при } \beta = \beta'; \end{cases}$$

оператор $\Pi(i, j)$ осуществляет перенос (без стирания) основного кода с i-й решетки на «пустую» решетку с номером j;

оператор $\Pi_f(i, j)$ переносит (со стиранием) заданный код числа f с i-й решетки на решетку j, располагая его через нулевой промежуток левее основного кода, который находится на j-й решетке.

В дальнейшем будут рассматриваться решетки с шагом 3 и 4. Ниже доказываются леммы для случая, когда берутся решетки с шагом 4 и специальных значений параметров i и j. Соответствующие утверждения для других случаев доказываются аналогично.

Лемма 8. *Пусть оператор* $p_>(\beta, \beta')$ *сравнивает числа* β *и* β' $(\beta \geqslant \beta')$, *расположенные соответственно на 1-й и 2-й решетках, причем начало кода* β' *находится в ячейке, лежащей непосредственно справа от ячейки, в которой начинается код* β. *Пусть, далее, в начальный момент головка обозревает начало кода* β, *а в заключительный — ту же самую ячейку. Наконец, пусть преобразование не меняет всей записи на ленте и завершается в состоянии* $\varkappa'$, *если* $p_> = 1$, *и в* $\varkappa''$, *если* $p_> = 0$. *Тогда существует машина Тьюринга, реализующая оператор* $p_>(\beta, \beta')$.

Таблица 18

	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$
0			$0R\varkappa_4$	$0R\varkappa_1$
1	$1R\varkappa_2$	$1R\varkappa_3$	$1R\varkappa_4$	$1R\varkappa_1$

Доказательство. Рассмотрим табл. 18. Очевидно, что эта машина останавливается при $\beta > \beta'$ в состоянии $\varkappa_2$ и при $\beta = \beta'$ в состоянии $\varkappa_1$. Для того чтобы построить искомую машину, необходимо вернуть головку данной машины в исходное положение. Лемма доказана.

Лемма 9. *Пусть* $\Pi(2, 3)$ *осуществляет перенос (без стирания) основного кода со 2-й решетки на «пустую» решетку с номером 3, причем в начальный момент обозревается левая единица основного кода на 2-й решетке, в конце преобразования — некоторая ячейка 2-й решетки, и запись на остальных решетках не меняется. Тогда мож-*

но построить машину Тьюринга, реализующую оператор $\Pi(2, 3)$.

Доказательство. Рассмотрим табл. 19. Она, очевидно, определяет машину, реализующую $\Pi(2, 3)$. После переноса машина останавливается на 2-й решетке правее основного кода в состоянии $\varkappa_9$. Лемма доказана.

Таблица 19

	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$	$\varkappa_8$	$\varkappa_9$
0 1	$1R\varkappa_2$	$1R\varkappa_3$	$0R\varkappa_4$ $1R\varkappa_4$	$0R\varkappa_5$ $1R\varkappa_5$	$0R\varkappa_6$ $1S\varkappa_1$	$0R\varkappa_7$	$0R\varkappa_8$ $1R\varkappa_8$	$0R\varkappa_9$ $1R\varkappa_9$	$1S\varkappa_1$

Лемма 10. *Пусть оператор* $\Pi_f(3, 1)$ *осуществляет перенос кода* f *(массив из единиц) с 3-й решетки на первую, располагая его через нулевой промежуток левее основного кода, причем в начальный момент обозревается левая единица основного кода, расположенного на 1-й решетке, в конце преобразования — левая единица построенного основного кода на 1-й решетке и код* f *на 3-й решетке стирается, а запись основного кода на 2-й решетке не меняется. Пусть, далее, на 4-й решетке в начальный момент находится массив из единиц, который захватывает все точки этой решетки в пределах кодов первых трех решеток (см. рис. 8), и в конце преобразования имеем на 4-й решетке массив из единиц, также захватывающий все точки решетки, но в пределах построенных кодов на первых трех решетках. Тогда можно построить машину Тьюринга, реализующую оператор* $\Pi_f(3, 1)$.

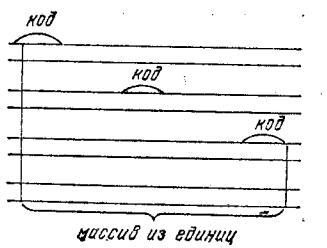


Рис. 8

Доказательство. 1) Оператор $\widetilde{\Phi}(a^i \to 1^i 0a)$ ставит на первой решетке слева от основного кода символы 10.

2) Смещаемся влево на единицу (оператор L_1), попадаем на 4-ю решетку.

3) Оператор $\tilde{L}_{IV}$ отыскивает левый конец массива из единиц на 4-й решетке.

4) Смещаемся влево на единицу (оператор L_1), попадаем на 3-ю решетку.

5) Оператор $\tilde{L}_{III}$ находит левый конец кода f на 3-й решетке (движение вправо).

6) Анализируем начало $a'a''$ кода на 3-й решетке, вычисляя предикат

$$\tilde{p}(a', a'') = \begin{cases} 0 & \text{при } a'' = 0, \\ 1 & \text{при } a'' = 1. \end{cases}$$

Если $\tilde{p} = 1$, то:

7) Стираем оператором $0_{III}(a')$ символ a'.

8) Смещаемся вправо на единицу (оператор R_1), попадаем на 4-ю решетку.

9) Оператор $\tilde{R}_{IV}$ находит правый конец массива из единиц на 4-й решетке.

10) Оператор R_1 осуществляет сдвиг на единицу вправо.

11) Двигаясь налево по первой решетке, находим (оператор $\tilde{L}_I$) левый конец кода на 1-й решетке.

12) Оператор $\tilde{\Phi}(a^1 \to 1^1 a)$ пристраивает к этому коду слева на 1-й решетке еще одну единицу, и, далее, возвращаемся к оператору 2).

Если $\tilde{p} = 0$, то:

13) Стираем оператором $0_{III}(a')$ символ a'.

14) Смещаемся вправо (оператор R_2) на две единицы — попадаем на 1-ю решетку.

15) Отыскиваем (оператор $\tilde{L}'_I$) левый конец основного кода на первой решетке и останавливаемся.

Считаем, что все моделирующие операторы на 4-й решетке в пределах рабочей зоны ставят единицы.

Операторная схема для $\Pi_f(3, 1)$ имеет следующий вид:

$$* \tilde{\Phi}(a^1 \to 1^1 0a) L_1 \tilde{L}_{II} L_1 \tilde{L}_{III} p(a'a'') \boxed{0_{III}(a') R_1 \tilde{R}_{II} R_1 \tilde{L}_I \tilde{\Phi}(a \to 1^1 a) p_0} 0_{III}(a') R_2 \qquad \tilde{L}'_I \omega$$

Лемма доказана.

Теорема 2. *Класс* $P_{выч}$ *замкнут относительно операции* Пр.

Доказательство. Пусть функция $f(x_1, \ldots, x_n, x_{n+1})$ определяется из вычислимых функций $\varphi(x_1, \ldots, x_n)$

и $\psi(x_1, \ldots, x_n, x_{n+1}, x_{n+2})$ при помощи операции Пр через схему

$$f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = \psi(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)).$$

Покажем, что $f(x_1, \ldots, x_n, x_{n+1}) \in P_{\text{выч}}$. Вместо данной схемы возьмем другую схему

$$f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = \psi'(f(x_1, \ldots, x_n, y), x_1, \ldots, x_n, y),$$

где $\psi'(x_{n+2}, x_1, \ldots, x_n, x_{n+1}) = \psi(x_1, \ldots, x_n, x_{n+1}, x_{n+2})$. В силу следствия 1 леммы 7 $\psi' \in P_{\text{выч}}$.

1) Оператор K_1 преобразует код$(\alpha_1, \ldots, \alpha_n, \beta)$ в четырехкратный код с буферным словом $U = 0001$, в результате чего получаем код, который разбивается при помощи четырех решеток с шагом 4 на три экземпляра кода$(\alpha_1, \ldots, \alpha_n, \beta)$, расположенных на первых трех решетках, и массива из единиц на 4-й решетке (см. рис. 9) в пределах кодов на первых трех решетках.

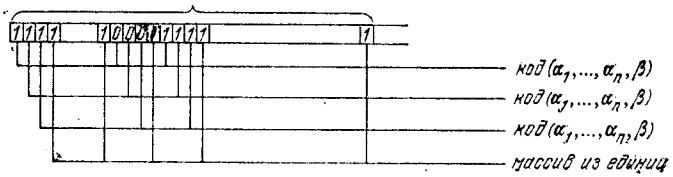


Рис. 9

2) Оператор Ф заменяет на второй решетке код β на код 0 и на третьей решетке стирает код β, останавливается над левой единицей третьей решетки, после чего на второй решетке имеем код $(\alpha_1, \ldots, \alpha_n, \beta')$ с $\beta' = 0$ и на третьей — код $(\alpha_1, \ldots, \alpha_n)$.

3) Оператор A_φ вычисляет на третьей решетке $\varphi(\alpha_1, \ldots, \alpha_n)$.

4) Оператор A_1 отыскивает начало кода β на первой решетке.

5) Оператор $p_>(\beta, \beta')$ $(\beta \geqslant \beta')$ производит сравнение кодов чисел β и β', расположенных на 1-й и 2-й решетках. В конце преобразования обозревается начало кода β.

Если $p_> = 1$, то:

6) Оператор L_1 отыскивает левую единицу кода на первой решетке.

7) Оператор $\Pi_f(3, 1)$ переносит код $f(\alpha_1, \ldots, \alpha_n, \beta')$, равный коду $\varphi(\alpha_1, \ldots, \alpha_n)$ при $\beta' = 0$, с 3-й решетки на первую, располагая его через нулевой зазор левее основного кода и стирая код f на 3-й решетке и в конце преобразования обозревает левую единицу построенного основного кода — кода $(f, \alpha_1, \ldots, \alpha_n, \beta)$ — на 1-й решетке.

8) Оператор L_{II} отыскивает левую единицу на 2-й решетке.

9) Оператор $\Pi(2, 3)$ переносит основной код со 2-й решетки на 3-ю (пустую) решетку, заканчивает работу в некоторой ячейке второй решетки.

10) Оператор L_{III} отыскивает левую единицу кода на 3-й решетке.

11) Оператор $\Pi_f(1,3)$ переносит код f с 1-й решетки на 3-ю, располагая его через нулевой зазор левее основного кода (вариант леммы), после чего на 3-й решетке получим код $(f, \alpha_1, \ldots, \alpha_n, \beta')$. В конце преобразования обозревается левая единица этого кода.

12) Оператор $\tilde{A}_{\psi'}$ вычисляет код $\psi'(f, \alpha_1, \ldots, \alpha_n, \beta')$ на 3-й решетке. Тем самым мы получаем

$$f(\alpha_1, \ldots, \alpha_n, \beta' + 1) = \psi'(f, \alpha_1, \ldots, \alpha_n, \beta').$$

13) Оператор $F_+(\beta', 1)$ увеличивает на единицу код β' на второй решетке и останавливается в некоторой ячейке 3-й решетки, после чего переходим к оператору 4).

Если $p_> = 0$, то:

14) Оператор $0(1, 2, 4)$ производит очистку решеток 1, 2, 4, ориентируясь массивом из единиц на 4-й решетке, и останавливается над левой единицей на 3-й решетке.

15) Оператор K_3 преобразует квазиосновной код, в котором все буферные слова пустые, в основной код. Мы получаем код $f(\alpha_1, \ldots, \alpha_n, \beta)$.

Здесь все операторы 1)—13) выбираются таким образом, что они не изменяют записи на других решетках, а на 4-й решетке в пределах рабочей зоны добавляют единицы.

Операторная схема имеет следующий вид:

$$*K_1 \Phi \tilde{A}_\varphi A_1 p_> (\beta, \beta') \boxed{L_I \Pi_f(3,1) L_{II} \Pi(2,3) L_{III} \Pi_f(1,3) \tilde{A}_{\psi'} F_+(\beta', 1) p_0} 0(1,2,4) K_3 \omega$$

Теорема доказана.

Замечание. В доказательстве теоремы по существу используется свойство области определения функции $f(x_1, \ldots, x_{n+1})$: если $f(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$ не определено, то при $\beta \geqslant \alpha_{n+1}$ не определено $f(\alpha_1, \ldots, \alpha_n, \beta)$.

Теорема 3. *Класс* $P_{\text{выч}}$ *замкнут относительно операции* μ.

Доказательство. Пусть $\varphi(x_1, \ldots, x_{n-1}, x_n)$ — вычислимая функция. Покажем, что функция $f(x_1, \ldots, x_n)$, где

$$f(x_1, \ldots, x_n) = \mu_y(\varphi(x_1, \ldots, x_{n-1}, y) = x_n),$$

вычислима.

Рассмотрим вспомогательную функцию

$$\varphi'(y, x_1, \ldots, x_{n-1}) = \varphi(x_1, \ldots, x_{n-1}, y).$$

В силу следствия 1 леммы 7 φ' вычислима.

1) Оператор K_1 преобразует код$(\alpha_1, \ldots, \alpha_n)$ в трехкратный код с буферным словом $U = 001$, в результате чего получаем код, который разбивается при помощи трех решеток с шагом 3 на два экземпляра кода$(\alpha_1, \ldots, \alpha_n)$, расположенных на первых двух решетках, и массива из единиц на 3-й решетке в пределах кодов первых двух решеток.

2) Оператор $\widetilde{\Phi}_{\text{I, II}}(a^1 \to 1^1 0a)$ пристраивает на первой и второй решетках к основным кодам слева код 0.

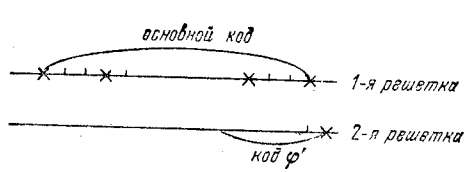


Рис. 10

3) Оператор $0_{\text{II}}(\alpha_n)$ стирает на второй решетке последний массив — код α_n.

4) Оператор $\widetilde{A}_{\varphi'}$ вычисляет на второй решетке $\varphi'(0, \alpha_1, \ldots, \alpha_{n-1})$.

5) Оператор A_1 «выравнивает» коды на 1-й и 2-й решетках, т. е. располагает их на решетках так, чтобы правый конец кода φ' находился в ячейке, непосредственно следующей за ячейкой, в которой расположен правый конец основного кода 1-й решетки (см. рис. 10).

Выравнивание кодов может быть осуществлено следующим образом: двигаясь по третьей решетке, выходим на

правый конец массива, затем отступаем вправо на 9 ячеек и, сдвигаясь влево на одну ячейку, попадаем на 2-ю решетку, в которую сдвигаем массив из единиц (код φ') (см. пример 5) и, далее, из правого конца сдвинутого кода φ' отступаем влево на одну ячейку — мы попадаем на 1-ю решетку, в которую переносим основной код (обобщение примера 5).

6) Оператор $p_{\neq}(\varphi', \alpha_n)$ сравнивает код φ' и код α_n, расположенные на 1-й и 2-й решетках (используем программу, аналогичную программе в лемме 8).

Если $p = 1$, т. е. $\varphi' \neq \alpha_n$, то:

7) Оператор $0(2)$ очищает 2-ю решетку.

8) Оператор $\tilde{F}_+(y, 1)$ увеличивает код y на 1-й решетке на единицу.

9) Оператор $\Pi(1, 2)$ переносит код с 1-й решетки на 2-ю и возвращается к оператору 3).

Если $p = 0$, т. е. $\varphi' = \alpha_n$, то:

10) Оператор $0(2, 3)$ очищает решетки 2 и 3.

11) Оператор $0(1)$ стирает на 1-й решетке все, кроме первого, массивы из единиц.

12) Оператор K_3 преобразует квазиосновной код в основной код и останавливается.

Операторная схема имеет вид

$$*K_1\tilde{\Phi}_{I,II}(a\overset{\downarrow}{-}1\overset{\downarrow}{0}a)\,0_{II}(\alpha_n)\tilde{A}_{\varphi'}A_1p_{\neq}(\varphi'\alpha_n)\boxed{0(2)\tilde{F}_+(y,1)\Pi(1,2)p_0}\,0(2,3)0(1)K_3\omega$$

При реализации отдельных операторов необходимо учитывать их согласование и простановку единиц на 3-й решетке в пределах рабочей зоны. Теорема доказана.

Замечание. В доказательстве используется по существу тот факт, что если хоть одно из значений $\varphi(\alpha_1, \ldots, \alpha_{n-1}, 0), \ldots, \varphi(\alpha_1, \ldots, \alpha_{n-1}, \mu_y - 1)$ не определено, то $f(\alpha_1, \ldots, \alpha_n)$ также не определено.

Теорема 4. *Класс* $P_{\text{выч}}$ *замкнут относительно системы операций* $R = \{C, \text{Пр}, \mu\}$.

Следствие. $P_{\text{пр}} \subseteq P_{\text{выч}}$.

Данная теорема дает возможность устанавливать вычислимость функций, не прибегая к построению машин Тьюринга, путем доказательства их частичной рекурсивности.

Примеры: а) На стр. 149 построен ряд примитивно-рекурсивных функций. В силу доказанного они являются также и вычислимыми.

б) Пусть $f(x)=x^2$. Очевидно, что $f\in P_{\text{пр}}$, так как

$$f(0)=0,\quad f(x+1)=f(x)+2x+1.$$

Следовательно, $f\in P_{\text{выч}}$.

§ 7. Формула Клини. Частичная рекурсивность вычислимых функций. Примеры полных систем

Этот параграф мы начнем с установления примитивной рекурсивности некоторых функций.

1. Пусть $\rho(x)$ обозначает число разрядов, содержащихся в двоичной записи числа x. Очевидно, что

$$\rho(0)=1,$$
$$\rho(x+1)=\rho(x)+\overline{\operatorname{Sg}}(2^{\rho(x)}\div(x+1)).$$

Следовательно, $\rho(x)\in P_{\text{пр}}$.

2. Пусть $\vartheta_n(x_1,\ldots,x_n)$ обозначает натуральное число, двоичная запись которого имеет вид

$$\underbrace{1\ldots1}_{x_1+1}\ 0\ \underbrace{1\ldots1}_{x_2+1}\ \ldots 0\ \underbrace{1\ldots1}_{x_n+1}$$

Примитивная рекурсивность доказывается индукцией по n:

$$\vartheta_1(0)=1,$$
$$\vartheta_1(x_1+1)=2\vartheta_1(x_1)+1,$$
$$\vartheta_n(x_1,\ldots,x_{n-1},0)=4\vartheta_{n-1}(x_1,\ldots,x_{n-1})+1,$$
$$\vartheta_n(x_1,\ldots,x_{n-1},x_n+1)=2\vartheta_n(x_1,\ldots,x_{n-1},x_n)+1.$$

3. Рассмотрим функцию $\pi(x_1,x_2)$ (см. табл. 20). Эта функция называется *пеановской функцией* и служит для нумерации всех пар (α_1,α_2) чисел из расширенного натурального ряда. Очевидно,

$$\pi(x_1,x_2)=\frac{(x_1+x_2)(x_1+x_2+1)}{2}+x_1=$$
$$=\left[\frac{(x_1+x_2)(x_1+x_2+1)}{2}\right]+x_1,$$

т. е. является примитивно-рекурсивной.

4. С пеановской функцией $\pi(x_1,x_2)$ связаны $\lambda(x)$ и $\mu(x)$. Пусть $\lambda(x)=\pi(0,x)=\left[\frac{x(x+1)}{2}\right]$, т. е. функция $\lambda(x)$

задается первой строкой табл. 20 для π. Значит, $\lambda(x) \in P_{\text{пр}}$.

Определим функцию $\mu(x)$ через табл. 21. Из таблицы видна связь функций π, λ и μ: значения функции π пробегают расширенный натуральный ряд и они делятся

Таблица 20

x_1 \ x_2	0	1	2	3	...
0	0	1	3	6	...
1	2	4	7	...	
2	5	8	...		
3	9	...			
...	...				

диагоналями $(x_1 + x_2 = c)$ на конечные куски; если $\alpha\,(\alpha \in E_{\aleph_0})$ — произвольное значение из таблицы для π, то $\mu(\alpha)$ указывает номер c диагонали, на которой находится α, а $\lambda(\mu(\alpha))$ — наименьшее число из $E_{\aleph_0}$, лежащее на этой диагонали.

Таблица 21

x	0	1	2	3	4	5	6	7	8	9	...
$\mu(x)$	0	1	1	2	2	2	3	3	3	3	...

Из этих соображений имеем

$$\mu(0) = 0,$$

$$\mu(x+1) = \mu(x) + \overline{\operatorname{Sg}}\{\mu(x) \dot{-} (x \dot{-} \lambda(\mu(x)))\}.$$

Поэтому $\mu(x) \in P_{\text{пр}}$.

5. Пусть (α_1, α_2) — произвольная пара. Тогда $\gamma = \pi(\alpha_1, \alpha_2)$ — ее номер. Обозначим через $l(x)$ и $r(x)$ функции, которые по номеру γ пары (α_1, α_2) дают ее

компоненты α_1 и α_2. Таким образом, $l(\gamma)=\alpha_1$ и $r(\gamma)=\alpha_2$. Данные функции удовлетворяют тождествам

$$\pi(l(x),\ r(x))\equiv x,\quad l(\pi(x_1,\ x_2))\equiv x_1,\quad r(\pi(x_1,\ x_2))\equiv x_2.$$

Так как

$$l(x)=x\dot{-}\lambda(\mu(x)),\quad r(x)=\mu(x)\dot{-}l(x),$$

то $l(x),\ r(x)\in P_{\text{пр}}$.

6. Функции π, l и r могут быть обобщены на случай многих переменных. Пеановская функция $\pi_s(x_1,\ \ldots,\ x_s)$ определяет номер s-ки $(\alpha_1,\ \ldots,\ \alpha_s)$ чисел из расширенного натурального ряда, а функции $t_1(x),\ \ldots,\ t_s(x)$ указывают по номеру s-ки значения ее компонент, т. е. числа $\alpha_1,\ \ldots,\ \alpha_s$.

Для $s=3$ данные функции определяются так:

$$\pi_3(x_1,\ x_2,\ x_3)=\pi(x_1,\ \pi(x_2,\ x_3)),$$

$$t_1(x)=l(x),\quad t_2(x)=l(r(x)),\quad t_3(x)=r(r(x)).$$

Очевидно, что

$$\pi_3(t_1(x),\ t_2(x),\ t_3(x))\equiv x,\quad t_1(\pi_3(x_1,\ x_2,\ x_3))\equiv x_1,$$

$$t_2(\pi_3(x_1,\ x_2,\ x_3))\equiv x_2,\quad t_3(\pi_3(x_1,\ x_2,\ x_3))\equiv x_3.$$

Из определения следует, что $\pi_3(x_1,\ x_2,\ x_3)$, $t_1(x)$, $t_2(x)$, $t_3(x)$ принадлежат $P_{\text{пр}}$.

Для дальнейших рассмотрений нам понадобится один способ построения функций.

Пусть $x=(x_1,\ \ldots,\ x_n)$ и функции $f_1(x,\ y),\ \ldots,\ f_s(x,\ y)$ заданы при помощи схемы одновременной примитивной рекурсии:

$$f_1(x,\ 0)=\varphi_1(x),$$

$$\ldots\ldots$$

$$f_s(x,\ 0)=\varphi_s(x),$$

$$f_1(x,\ y+1)=\psi_1(x,\ y,\ f_1(x,\ y),\ \ldots,\ f_s(x,\ y)),$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$f_s(x,\ y+1)=\psi_s(x,\ y,\ f_1(x,\ y),\ \ldots,\ f_s(x,\ y)),$$

представляющей естественное обобщение схемы примитивной рекурсии. Эту схему используем для случая, когда $\varphi_1,\ \ldots,\ \varphi_s,\ \psi_1,\ \ldots,\ \psi_s$ примитивно-рекурсивны. Покажем, что эти функции могут быть получены из функций $\varphi_1,\ \ldots,\ \varphi_s,\ \psi_1,\ \ldots,\ \psi_s$, а также примитивно-рекурсивных

функций π_s, t_1, ..., t_s при помощи суперпозиций и примитивной рекурсии.

Рассмотрим функцию

$$f(x, y) = \pi_s(f_1(x, y), \ldots, f_s(x, y));$$

мы имеем

$$f(x, 0) = \pi_s(f_1(x, 0), \ldots, f_s(x, 0)) = \pi_s(\varphi_1(x), \ldots, \varphi_s(x)),$$

$$f(x, y+1) = \pi_s(f_1(x, y+1), \ldots, f_s(x, y+1)) =$$

$$= \pi_s(\psi_1(x, y, f_1(x, y), \ldots, f_s(x, y)), \ldots$$

$$\ldots, \psi_s(x, y, f_1(x, y), \ldots, f_s(x, y))) =$$

$$= \pi_s(\psi_1(x, y, t_1(f(x, y)), \ldots, t_s(f(x, y))), \ldots$$

$$\ldots, \psi_s(x, y, t_1(f(x, y)), \ldots, t_s(f(x, y)))) = \psi(x, y, f(x, y)),$$

где

$$\psi(x, y, z) = \pi_s(\psi_1(x, y, t_1(z), \ldots, t_s(z)), \ldots$$

$$\ldots, \psi_s(x, y, t_1(z), \ldots, t_s(z))).$$

Значит, $f(x, y)$ может быть получена при помощи примитивной рекурсии из $\pi_s(\varphi_1, \ldots, \varphi_s)$ и ψ, т. е. $f(x, y) \in P_{\text{пр}}$. Далее,

$$f_1(x, y) = t_1(f(x, y)), \ldots, f_s(x, y) = t_s(f(x, y)),$$

поэтому $f_1(x, y), \ldots, f_s(x, y) \in P_{\text{пр}}$.

Докажем теперь теорему о представлении вычислимой функции (а значит и произвольной частично-рекурсивной функции) через примитивно-рекурсивные функции в специальной форме (аналог теоремы Клини).

Теорема 5. *Для всякой вычислимой функции* $f(x_1, \ldots, x_n)$ *существуют такие примитивно-рекурсивные функции* $F_f(x_1, \ldots, x_n, y)$ *и* $G_f(x_1, \ldots, x_n, y)$, *что*

$$f(x_1, \ldots, x_n) = F_f(x_1, \ldots, x_n, \mu_y(G_f(x_1, \ldots, x_n, y) = 0)).$$

Доказательство. Пусть $f(x_1, \ldots, x_n)$ — произвольная вычислимая функция, которую мы будем записывать короче как $f(x)$, где $x = (x_1, \ldots, x_n)$. Рассмотрим машину Тьюринга, вычисляющую ее правильным образом. Пусть T — программа машины и $\varkappa_1, \ldots, \varkappa_r$ — состояния машины. Введем новое состояние $\varkappa_0$ (символ $\varkappa_0$ отличен от $\varkappa_1, \ldots, \varkappa_r$) и дополним программу T, заполнив все ее пустые клетки, а также клетки присоединенного столбца $\varkappa_0$ следующим образом: если пустая клетка принадлежит строке a, то в нее помещается команда $aS\varkappa_0$. Полученную программу обозначим через T'. Если усло-

виться, что машина, соответствующая T' (которая фактически работает так же, как исходная), останавливается, попадая в состояние $\varkappa_0$, то она будет вычислять функцию f так же, как исходная машина с программой T.

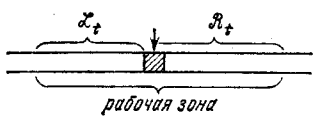


Рис. 11

Для машины T' рассмотрим в момент времени t ее ленту и отметим на ней рабочую зону, т. е. совокупность всех ячеек ленты, состоящую из ячеек, в которых побывала головка машины, и ячеек, в которых записан исходный основной код для x. По отношению к ячейке, которую в момент t обозревает головка, рабочая зона разбивается на две части $\mathscr{L}_t$ и $\mathscr{R}_t$ — левую и правую части (см. рис. 11). Кусок $\mathscr{L}_t$ расположен левее ячейки, обозреваемой головкой в момент t, $\mathscr{R}_t$ — содержит остальные ячейки рабочей зоны. Обозначим через f_2 — натуральное число, запись которого в двоичном счислении находится в ячейках $\mathscr{L}_t$, и через f_1 — натуральное число, запись которого в двоичном счислении находится в ячейках $\mathscr{R}_t$, если ее читать справа налево (инверсным образом). Очевидно, что

$$f_1 = f_1(x', t), \quad f_2 = f_2(x', t),$$

где x' — натуральное число, запись которого в двоичном счислении совпадает с записью исходного кода x, читаемого справа налево.

Пусть $f_3(x', t)$ — номер состояния в момент времени t, если в начальный момент запись на ленте характеризуется натуральным числом x'.

В момент времени $t = 0$ будет $\mathscr{L}_t = \Lambda$ (пусто), $\mathscr{R}_t$ совпадает с множеством ячеек, занятых записью основного кода. Поэтому

$$f_1(x', 0) = x',$$
$$f_2(x', 0) = 0,$$
$$f_3(x', 0) = 1.$$

С другой стороны, очевидно,

$$f_1(x', t+1) = \psi_1(f_1(x', t), f_2(x', t), f_3(x', t)),$$
$$f_2(x', t+1) = \psi_2(f_1(x', t), f_2(x', t), f_3(x', t)),$$
$$f_3(x', t+1) = \psi_3(f_1(x', t), f_2(x', t), f_3(x', t)).$$

В самом деле, зная в момент времени t числа $f_1(x', t)$, $f_2(x', t)$, $f_3(x', t)$, мы находим число, обозреваемое головкой в момент времени t,— это будет младший разряд в двоичной записи $f_1(x', t)$ и состояние машины, номер которого есть $f_3(x', t)$. Эти две величины позволяют по таблице T' найти новое значение этой ячейки, новое состояние (номер состояния) и характер движения и, в конечном счете, числа $f_1(x', t+1)$, $f_2(x', t+1)$, $f_3(x', t+1)$. Эти преобразования и дают формулы для ψ_1, ψ_2 и ψ_3. Сейчас мы найдем их явное выражение. Для этого обозначим через $T_1(z_1, z_2)$, $T_2(z_1, z_2)$ и $T_3(z_1, z_2)$ соответственно функции, определяемые таблицей и дающие по входному символу и номеру состояния соответственно новый символ, номер нового состояния и номер движения (2 — для движения L, 1 — для движения S и 0 — для движения R). Для остальных значений из расширенного натурального ряда полагаем значения этих функций равными 0. Очевидно, что T_1, T_2, $T_3 \in P_{\text{пр}}$, так как они в конечном числе точек принимают ненулевые значения.

Обозначим через $\chi(z)$ *) младший разряд z и для сокращения записи положим

$$f_1 = f_1(x', t),$$
$$f_2 = f_2(x', t),$$
$$f_3 = f_3(x', t),$$
$$\gamma = T_1(\chi(f_1), f_3),$$
$$d = T_3(\chi(f_1), f_3).$$

Тогда при любом $d = 0, 1, 2$

$$f_3(x', t+1) = T_2(\chi(f_1), f_3).$$

Соответствующие равенства для $f_1(x', t+1)$ и $f_2(x', t+1)$ составим сначала отдельно для каждого случая:

а) $d = 1$ (движение S)

$$f_1(x', t+1) = f_1 \dot{-} \chi(f_1) + \gamma,$$
$$f_2(x', t+1) = f_2;$$

*) Очевидно, что $\chi(z) \in P_{\text{пр}}$.

б) $d=0$ (движение R)

$$f_1(x', t+1) = \left[\frac{f_1}{2}\right],$$

$$f_2(x', t+1) = 2f_2 + \gamma;$$

в) $d=2$ (движение L)

$$f_1(x', t+1) = 2(f_1 \dot{-} \chi(f_1) + \gamma) + \chi(f_2),$$

$$f_2(x', t+1) = \left[\frac{f_2}{2}\right].$$

Вспоминая, что $f_1(x', t+1)$, $f_2(x', t+1)$ и $f_3(x', t+1)$ — это значения функций ψ_1, ψ_2 и ψ_3, и объединяя соответствующие равенства из разных случаев, имеем

$$\psi_1(f_1, f_2, f_3) = (f_1 \dot{-} \chi(f_1) + T_1(\chi(f_1), f_3)) \cdot d +$$

$$+ \left[\frac{f_1}{2}\right] \cdot \overline{\mathrm{Sg}}\, d + \chi(f_2) \cdot \mathrm{Sg}(d \dot{-} 1),$$

$$\psi_2(f_1, f_2, f_3) = (2f_2 + T_1(\chi(f_1), f_3)) \cdot \overline{\mathrm{Sg}}\, d +$$

$$+ f_2 \cdot \mathrm{Sg}\, d \cdot \overline{\mathrm{Sg}}(d \dot{-} 1) + \left[\frac{f_2}{2}\right] \cdot \mathrm{Sg}(d \dot{-} 1),$$

$$\psi_3(f_1, f_2, f_3) = T_2(\chi(f_1), f_3).$$

Отсюда вытекает, что $\psi_1, \psi_2, \psi_3 \in P_{\text{пр}}$.

Для функций f_1, f_2 и f_3 мы имеем схему одновременной примитивной рекурсии. Следовательно, мы можем утверждать, что $f_1, f_2, f_3 \in P_{\text{пр}}$. Возьмем теперь

$$\mu_t\{f_3(x', t) = 0\}.$$

Данная функция принадлежит классу $P_{\text{чр}}$ и определяет для каждого x' момент останова машины. Если эту величину подставить в f_1, то получим

$$f_1(x', \mu_t\{f_3(x', t) = 0\}),$$

т. е. если машина останавливается, то получим натуральное число, двоичной записью которого будет код $f(x)$.

В таком случае, если учесть, что

$$x' = \vartheta'(x_1, \ldots, x_n),$$

где $\vartheta'(x_1, \ldots, x_n) = \vartheta_n(x_n, \ldots, x_1)$, то

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

$$= \rho(f_1(\vartheta'(x_1, \ldots, x_n), \mu_t(f_3(\vartheta'(x_1, \ldots, x_n), t) = 0))) \dot{-} 1.$$

Если положить

$$F_f(x_1, \ldots, x_n, y) = \rho(f_1(\vartheta'(x_1, \ldots, x_n), y)) \dot{-} 1,$$

$$G_f(x_1, \ldots, x_n, y) = f_3(\vartheta'(x_1, \ldots, x_n), y),$$

то получим требуемое. Теорема доказана.

Как следствие из теорем получается

Теорема 6. $P_{\text{выч}} = P_{\text{чр}}$.

Таблица 22

x_1	x_2				
	0	1	2	3	...
0	1	0	0	0	...
1	0	2	0	0	...
2	0	0	3	0	...
3	0	0	0	4	...
...	...	...	...	...	...

Из последних двух теорем имеем также, что каждая частично-рекурсивная функция может быть записана через примитивно-рекурсивные функции в виде канонического уравнения, даваемого представлением Клини.

Теорема 7. *Система функций* $\{0, S(x), I_1^1(x)\}$ *полна в* $P_{\text{выч}}$ *относительно набора операций* $\{C, \text{Пр}, \mu\}$.

Теорема 8. *Система функций* $\{0, S(x)\}$ *полна в* $P_{\text{выч}}$ *относительно набора операций* $\{C, \text{Пр}, \mu\}$.

Доказательство. Функция $I_1^1(x)$ определяется через 0 и $S(x)$ при помощи следующей схемы:

$$I_1^1(0) = 0,$$

$$I_1^1(x+1) = S(I_1^1(x)).$$

Теорема 9. *В функциональной системе* $(P'_{\text{выч}}, C, \text{Пр}, \mu)$ *существует аналог функции Шеффера**).

Доказательство. Возьмем функцию $f(x_1, x_2)$ (см. табл. 22). Очевидно, $f(x_1, x_1) = S(x_1)$ и $f(x_1, S(x_1)) = 0(x_1)$. Затем, как в предыдущей теореме, получаем $I_1^2(x_1, x_2)$, $I_1^1(x_1) = I_1^2(x_1, x_1)$ и все $I_m^n(x_1, \ldots, x_n)$. Отсюда легко построить также класс $P'_{\text{выч}}$.

---

*) Здесь через $P'_{\text{выч}}$ обозначим множество $P_{\text{выч}} \setminus E_{\aleph_0}$.

ЧАСТЬ II

КОМБИНАТОРНЫЙ АНАЛИЗ

Комбинаторный анализ занимается изучением объектов из конечного множества $E = \{a_1, \ldots, a_n\}$ и их свойств. Этими объектами могут быть подмножества множества E, подмножества с повторяющимися элементами из множества E, упорядоченные подмножества множества E и т. п.

Комбинаторный анализ является разделом дискретной математики, истоки которого уходят в глубокую древность. В настоящее время интерес к нему значительно усилился. Благодаря этому комбинаторный анализ сегодня превратился в достаточно развитую ветвь математики, которая непрерывно разрастается. Это делает трудным четко очертить круг объектов и их свойств, которые принадлежат комбинаторике. Ввиду этого мы начинаем с описания простейших (элементарных) комбинаторных объектов.

§ 1. Комбинаторные объекты и комбинаторные числа

1. Система подмножеств множества E. Пусть $E = \{a_1, \ldots, a_n\}$ — конечное множество. Рассматриваются все его подмножества. Эту систему обозначим через $\mathfrak{E}_n$.

Пример. $E = \{a_1, a_2, a_3\}$. Система его подмножеств имеет вид:

$$\mathfrak{E}_3 = \{\Lambda, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}.$$

2. Размещения элементов из E по k. Пусть $E = \{a_1, \ldots, a_n\}$. *Размещением* элементов из E по k называется упорядоченное подмножество из k элементов, принадлежащих E.

Пример. $E = \{a_1, a_2, a_3\}$ и $k = 2$. Выпишем все размещения из этого множества по 2:

$$(a_1, a_2), (a_2, a_1), (a_1, a_3), (a_3, a_1), (a_2, a_3), (a_3, a_2).$$

3. Перестановки элементов множества E. Пусть $E = \{a_1, \ldots, a_n\}$. *Перестановками* называются упорядоченные подмножества из n элементов множества E.

Пример. $E=\{a_1, a_2, a_3\}$. Перестановками множества E будут (a_1, a_2, a_3), (a_1, a_3, a_2), (a_2, a_1, a_3), (a_2, a_3, a_1), (a_3, a_1, a_2), (a_3, a_2, a_1).

Очевидно, что перестановки — частный случай размещений элементов из E по k, когда $k=n$.

4. Сочетания элементов из E по k. *Сочетанием* элементов из E по k называется неупорядоченное подмножество из k элементов, принадлежащих E.

Пример. $E=\{a_1, a_2, a_3\}$ и $k=2$. Сочетаниями из E по 2 будут $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$.

5. Сочетания с повторениями элементов из E по k. *Сочетанием с повторениями* элементов из E по k является неупорядоченная система из k элементов, принадлежащих E, в которой допускается повторение элементов.

Пример. $E=\{a_1, a_2, a_3\}$ и $k=2$. Сочетаниями с повторениями из E по 2 будут $\{a_1, a_1\}$, $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_2\}$, $\{a_2, a_3\}$, $\{a_3, a_3\}$.

6. n-мерный куб размера k ($k\geqslant 2$). Совокупность всех наборов $(\alpha_1, \ldots, \alpha_n)$ (упорядоченных сочетаний с повторениями) из множества $E_k=\{0, 1, \ldots, k-1\}$ по n называется *n-мерным кубом размера k* и обозначается через E_k^n $\left(E_k^n=\underbrace{E_k\times\ldots\times E_k}_{n\text{ раз}}\right)$.

Пример 1. 3-мерный куб размера 2. $E_2=\{0, 1\}$. Мы имеем следующую совокупность наборов: (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1). Эти наборы можно рассматривать как вершины единичного 3-мерного куба (см. рис. 1).

Пример 2. На рис. 2 изображен 3-мерный куб размера 3.

7. Разбиения множества E. *Разбиением* множества E называется неупорядоченная система из непустых подмножеств $(\mathscr{E}_1, \mathscr{E}_2, \ldots, \mathscr{E}_k)$ множества E, обладающая двумя свойствами:

1) $\mathscr{E}_1\cup\ldots\cup\mathscr{E}_k=E$;

2) для любых i, j $(i\neq j)$ множества $\mathscr{E}_i$ и $\mathscr{E}_j$ не пересекаются, т. е. $\mathscr{E}_i\cap\mathscr{E}_j=\Lambda$.

Пример. $E=\{a_1, a_2, a_3\}$. Разбиениями будут $\{\{a_1, a_2, a_3\}\}$, $\{\{a_1\}, \{a_2, a_3\}\}$, $\{\{a_2\}, \{a_1, a_3\}\}$, $\{\{a_3\}, \{a_1, a_2\}\}$, $\{\{a_1\}, \{a_2\}, \{a_3\}\}$.

Существует много других типов комбинаторных объектов. Например: покрытия конечного множества, блок-схемы, булевы функции, системы частично упорядоченных множеств и т. п. Сведения о них можно найти

в [29, 30, 32]. Во многих из них комбинаторная сторона играет не основную роль (например, булевы функции), потому их естественно включать в другие разделы дискретной математики.

Наряду с классами комбинаторных объектов рассматриваются и

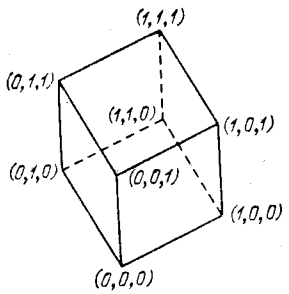


Рис. 1

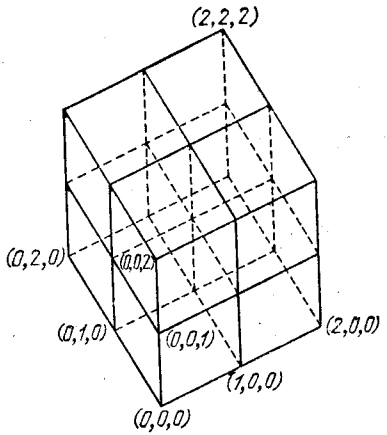


Рис. 2

так называемые комбинаторные числа, характеризующие число объектов в данном классе и зависящие от некоторых параметров.

§ 2. Простейшие свойства комбинаторных объектов и чисел

Здесь изучаются свойства, которые легко усматриваются из «комбинаторных» соображений.

1. Подмножества множества $E = \{a_1, a_2, \ldots, a_n\}$. В качестве комбинаторного числа, связанного с $\mathfrak{E}_n$, обычно берут мощность $\mathfrak{E}_n$, т. е. величину $|\mathfrak{E}_n|$. Пусть $\mathscr{E} \in \mathfrak{E}_n$. Сопоставим $\mathscr{E}$ взаимнооднозначным образом двоичный набор $(\alpha_1, \alpha_2, \ldots, \alpha_n)$:

$$\alpha_i = \begin{cases} 1, & \text{если } a_i \in \mathscr{E}, \\ 0, & \text{если } a_i \notin \mathscr{E}. \end{cases}$$

Отсюда получаем, что

$$|\mathfrak{E}_n| = |E_2^n| = 2^n.$$

С другой стороны, отсюда же получается простой алго-

ритм порождения (перечисления) всех подмножеств. Для этого строим все наборы $(\alpha_1, \ldots, \alpha_n)$ исходя из $(0, \ldots, 0)$, на каждом шаге прибавляя 1 к соответствующему двоичному числу.

2. Размещения элементов из E по k. Обозначим число таких размещений через $(n)_k$. При построении конкретного размещения на 1-е место можно поставить любой из n элементов, на 2-е место — любой из $n-1$ оставшихся элементов и т. д. Поэтому

$$(n)_k = n(n-1)\ldots(n-k+1) \text{ при } 1 \leqslant k \leqslant n. \quad (1)$$

Считаем

$$(n)_k = 0 \quad \text{при} \quad k > n,$$

поскольку при $k > n$ не существует размещений из n по k. Кроме того, полагаем

$$(0)_0 = (n)_0 = 1.$$

Для чисел $(n)_k$ выполняются тождества

$$(n)_k = n(n-1)_{k-1},$$

$$\ldots\ldots\ldots (n)_k = (n)_{k-1} \cdot (n-k+1).$$

Используя первое из них, с линейной сложностью строим таблицу 1.

3. Перестановки элементов множества E. Перестановка из элементов — частный случай размещения при $k = n$. Поэтому для числа перестановок имеем

$$(n)_n = n(n-1)\ldots 2 \cdot 1 = n!$$

Как обычно, считаем $0! = 1$. Числа $n!$ в табл. 1 расположены по диагонали.

Далее мы приводим сведения о числе e и неравенствах, связанных с числом e, а также оценки для $n!$

Число e. В дальнейшем это число будет часто встречаться. Дадим его определение. Покажем, что существует $\lim\limits_{n\to\infty}\left(1+\frac{1}{n}\right)^n$. Пусть $a_n = \left(1+\frac{1}{n}\right)^n$. Очевидно,

$$a_n = 1 + \binom{n}{1}\frac{1}{n} + \binom{n}{2}\frac{1}{n^2} + \ldots + \binom{n}{n}\frac{1}{n^n} =$$

$$= 1 + 1 + \left(1-\frac{1}{n}\right)\frac{1}{1\cdot 2} + \ldots + \left(1-\frac{1}{n}\right)\ldots\left(1-\frac{n-1}{n}\right)\frac{1}{1\cdot 2\cdot\ldots\cdot n}.$$

Таблица 1

n	k										
	0	1	2	3	4	5	6	7	8	9	...
0	1										...
1	1	1									...
2	1	2	2					0			...
3	1	3	6	6							...
4	1	4	12	24	24						...
5	1	5	20	60	120	120					...
6	1	6	30	120	360	720	720				...
7	1	7	42	210	840	2 520	5 040	5 040			...
8	1	8	56	336	1680	6 720	20 160	40 320	40 320		...
9	1	9	72	504	3024	15 120	60 480	181 440	362 880	362 880	...
...	...	...	...	...	...	...	...	...	...	...	...

Сравним это число с a_{n+1}:

$$a_{n+1} = 1 + 1 + \left(1 - \frac{1}{n+1}\right)\frac{1}{1\cdot 2} + \ldots + \left(1 - \frac{1}{n+1}\right)\cdots$$

$$\ldots\left(1 - \frac{n-1}{n+1}\right)\frac{1}{1\cdot 2\cdot\ldots\cdot n} + \left(1 - \frac{1}{n+1}\right)\cdots\left(1 - \frac{n}{n+1}\right)\frac{1}{1\cdot 2\cdot\ldots\cdot(n+1)}.$$

Члены, входящие в a_{n+1}, соответственно не меньше членов из a_n.

Отсюда $a_n < a_{n+1}$ и $\{a_n\}$ — монотонно возрастающая последовательность. Кроме того, так как

$$a_n = 1 + 1 + \left(1 - \frac{1}{n}\right)\frac{1}{1\cdot 2} + \ldots + \left(1 - \frac{1}{n}\right)\cdots$$

$$\ldots\left(1 - \frac{n-1}{n}\right)\frac{1}{1\cdot 2\cdot\ldots\cdot n} < 1 + 1 + \frac{1}{2} + \frac{1}{2^2} + \ldots < 3,$$

то эта последовательность ограничена сверху. Поэтому она имеет предел — его и обозначают через e, т. е.

$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n.$$

Из доказательства следует, что при $n \geqslant 1$

$$2 < \left(1+\frac{1}{n}\right)^n < e \leqslant 3 \tag{2}$$

и, в частности,

$$n \ln\left(1+\frac{1}{n}\right) < 1. \tag{3}$$

Для последующего важно и другое неравенство

$$\left(n+\frac{1}{2}\right)\ln\left(1+\frac{1}{n}\right) > 1, \tag{4}$$

которое может быть получено из рассмотрения графика функции $y = \frac{1}{x}$ (рис. 3). Сравнивая площади фигур, первая из которых ограничена осью x, прямыми $x = n$, $x = n+1$ и графиком $y = 1/x$, вторая — осью x, прямыми $x = n$, $x = n+1$ и касательной к кривой в точке $x = n + \frac{1}{2}$, имеем

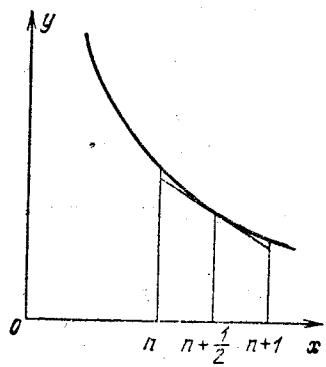


Рис. 3

$$\int_n^{n+1} \frac{dx}{x} > \frac{1}{n+1/2}$$

или

$$\left(n+\frac{1}{2}\right)\ln\left(1+\frac{1}{n}\right) > 1.$$

Оценки для $n!$ Приведем две грубые оценки для $n!$, использующие элементарные доказательства.

Первое неравенство

$$n! > (n/e)^n. \tag{5}$$

Доказываем по индукции. При $n=1$ имеем $1>1/e$. Индуктивный переход:

$$(n+1)!>(n+1)\left(\frac{n}{e}\right)^n=\frac{n+1}{e^n}n^n>\frac{n+1}{e^n}\frac{(n+1)^n}{e}=$$
$$=\left(\frac{n+1}{e}\right)^{n+1},$$

где использовано неравенство $n^n>(n+1)^n/e$ (см. (2)).

Второе неравенство

$$n!\leqslant 2\left(\frac{n}{2}\right)^n. \tag{6}$$

Доказательство использует хорошо известное неравенство $\sqrt{ab}\leqslant(a+b)/2$ (для положительных a и b):

$$n!=n\,[(1\cdot(n-1))(2\cdot(n-2))\ldots]\leqslant$$
$$\leqslant 2\left(\frac{n}{2}\right)\left[\left(\frac{n}{2}\right)^2\left(\frac{n}{2}\right)^2\ldots\right]=2\left(\frac{n}{2}\right)^n.$$

4. Сочетания элементов из E по k. Сочетание отличается от размещения тем, что в нем не учитывается порядок. Поэтому каждому сочетанию соответствует $k!$ размещений. Отсюда получается формула для числа $\binom{n}{k}$ сочетаний из n элементов по k $(0\leqslant k\leqslant n)$:

$$\binom{n}{k}=\frac{(n)_k}{k!}=\frac{n(n-1)\ldots(n-k+1)}{k!}=\frac{n!}{k!\,(n-k)!}. \tag{7}$$

Из данной формулы вытекает, что

$$\binom{0}{0}=\binom{n}{0}=\binom{n}{n}=1 \text{ и } \binom{n}{k}=\binom{n}{n-k}.$$

Иногда удобно выражение $\binom{n}{k}$ доопределить и для случая $k>n$:

$$\binom{n}{k}=0,$$

поскольку при $k>n$ не существует сочетаний из n по k. В дальнейшем, если не будет специальных оговорок, считаем, что $k\leqslant n$. Отметим одно тождество, которое легко получается из (7)

$$\binom{n}{i}\binom{i}{r}=\binom{n}{r}\binom{n-r}{i-r}, \tag{8}$$

если $0\leqslant r\leqslant i\leqslant n$.

По аналогии с понятием унимодальной функции [1] введем понятие унимодальной последовательности $\{a_k\}$, где $k = 0, 1, \ldots, n$.

Определение. Последовательность $\{a_k\}$ действительных чисел называется *унимодальной*, если существует такое k_n, что $a_0 < a_1 < \ldots < a_{k_n} \geqslant a_{k_n+1} > a_{k_n+2} > \ldots > a_n$, т. е.:

1) последовательность строго возрастает на отрезке $[0, k_n]$ при $k_n > 0$;

2) последовательность строго убывает на отрезке $[k_n + 1, n]$ при $k_n + 1 < n$;

3) максимальное значение принимается не более чем в двух точках: k_n и, быть может, $k_n + 1$ *).

Теорема 1. *Последовательность чисел* $\left\{\binom{n}{k}\right\}$, $k = 0, 1, \ldots, n$, *унимодальна, и* $k_n = [n/2]$. *При* четном n *максимум достигается в точке* $k_n = [n/2] = n/2$, *а при* нечетном n *— в двух точках:* $k_n = [n/2] = (n-1)/2$ *и* $k_n + 1 = (n+1)/2$.

Доказательство. Оценим отношение двух соседних членов последовательности $\binom{n}{k} \Big/ \binom{n}{k-1} = \frac{n-k+1}{k}$.

а) При $k \leqslant [n/2]$, т. е. $k < (n+1)/2$, имеем $(n-k+1)/k > 1$. Поэтому $\binom{n}{k} \Big/ \binom{n}{k-1} > 1$.

б) При $k - 1 \geqslant n - [n/2]$, т. е. $k - 1 > n - \frac{n+1}{2} = \frac{n-1}{2}$, имеем $\frac{n-k+1}{k} < 1$. Поэтому $\binom{n}{k} \Big/ \binom{n}{k-1} < 1$.

Теорема доказана.

Следствие. Максимальное значение $\binom{n}{k}$ при фиксированном n равно $\binom{n}{[n/2]}$.

Обозначим через $G_{n,k}$ множество всех сочетаний из $\{a_1, \ldots, a_n\}$ по k и через $G_{n-1,k}(G_{n-1,k-1})$ — множество всех сочетаний из $\{a_1, \ldots, a_{n-1}\}$ по k (соответственно по $k-1$). Так как каждому сочетанию из $G_{n,k}$, если оно содержит элемент a_n, соответствует сочетание из $G_{n-1,k-1}$, а если оно не содержит a_n, соответствует сочетание из

*) См. [1]. Легко усмотреть следующую связь: если существует $f(x)$, определенная на $[0, n]$ и $f(k) = a_k$, тогда из строгой унимодальности $f(x)$ следует унимодальность $\{a_k\}$.

$G_{n-1,\,k}$, то существует взаимно однозначное соответствие между

$$G_{n,\,k} \quad \text{и} \quad G_{n-1,\,k} \cup G_{n-1,\,k-1}.$$

В силу этого

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$

Данное рекуррентное соотношение позволяет (и при том с линейной сложностью) построить для чисел $\binom{n}{k}$ таблицу, называемую *треугольником Паскаля.*

Таблица 2

n	k										
	0	1	2	3	4	5	6	7	8	9	...
0	1										...
1	1	1									...
2	1	2	1								...
3	1	3	3	1							...
4	1	4	6	4	1						...
5	1	5	10	10	5	1					...
6	1	6	15	20	15	6	1				...
7	1	7	21	35	35	21	7	1			...
8	1	8	28	56	70	56	28	8	1		...
9	1	9	36	84	126	126	84	36	9	1	...
...	...	...	...	...	...	...	...	...	...	...	...

0

Сочетания элементов из $E = \{a_1, \ldots, a_n\}$ по k являются специальными подмножествами из E, содержащими ровно k элементов. Соответствующие им наборы $(\alpha_1, \ldots, \alpha_n)$ содержат ровно k единиц и образуют *k-й слой* n-мерного куба E_2^n. Отсюда следует, что k-й слой содержит $\binom{n}{k}$

точек. Следовательно, E_2^n разбивается в прямую сумму слоев с номерами $k = 0, 1, \ldots, n$ и

$$\binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{n} = 2^n. \tag{9}$$

С числами $\binom{n}{k}$ связано функциональное тождество, называемое *формулой для бинома Ньютона*:

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x + \ldots + \binom{n}{k}x^k + \ldots + \binom{n}{n}x^n. \tag{10}$$

В самом деле, коэффициент при x^k получается всевозможными выборами из k скобок $(1+x)$ переменного x и из остальных скобок 1, что дает как раз $\binom{n}{k}$ слагаемых.

Полагая в (10) $x = 1$, мы получаем тождество (9). Если в (10) взять $x = -1$, то получим

$$\binom{n}{0} - \binom{n}{1} + \ldots + (-1)^n\binom{n}{n} = 0. \tag{11}$$

Покажем, далее, что

$$\sum_{i=0}^{n}(-1)^{i-r}\binom{n}{i}\binom{i}{r} = \begin{cases} 1 \text{ при } r = n, \\ 0 \text{ при } r < n. \end{cases} \tag{12}$$

Опираясь на тождества (8) и (11), имеем

$$\sum_{i=0}^{n}(-1)^{i-r}\binom{n}{i}\binom{i}{r} = \sum_{i=r}^{n}(-1)^{i-r}\binom{n}{r}\binom{n-r}{i-r} =$$

$$= \binom{n}{r}\sum_{i=r}^{n}(-1)^{i-r}\binom{n-r}{i-r} = \begin{cases} 1 \text{ при } r = n, \\ 0 \text{ при } r < n. \end{cases}$$

Аналогично доказывается, что при $m \geqslant n$

$$\sum_{i=r}^{n}(-1)^{i-r}\binom{m-i}{m-n}\binom{m-r}{m-i} = \begin{cases} 1 \text{ при } r = n, \\ 0 \text{ при } r < n. \end{cases} \tag{13}$$

5. Сочетания с повторениями элементов из E по k. Обозначим через H_n^k число сочетаний с повторениями элементов из множества $E = \{a_1, \ldots, a_n\}$ по k.

Теорема 2. $H_n^k = \binom{n+k-1}{k}$.

Доказательство. Рассмотрим некоторое сочетание с повторениями из этого множества, содержащее k

объектов. Пусть в сочетании встречаются s элементов $a_{i_1}, \ldots, a_{i_s}$ $(1 \leqslant i_1 < \ldots < i_s \leqslant n)$ соответственно с кратностями $k_1, \ldots, k_s$, где $k_1 + \ldots + k_s = k$. Мы будем это сочетание записывать в следующем виде:

$$a_{i_1}^{k_1} a_{i_2}^{k_2} \ldots a_{i_s}^{k_s}.$$

Установим взаимно однозначное соответствие между сочетаниями с повторениями объектов множества $\{a_1, \ldots, a_n\}$ по k и обычными сочетаниями из множества $\{a_1, \ldots, a_n, b_1, \ldots, b_{k-1}\}$ по k, где символы $a_1, \ldots, a_n, b_1, \ldots, b_{k-1}$ попарно различны. Для этого сочетанию с повторениями

$$a_{i_1}^{k_1} a_{i_2}^{k_2} \ldots a_{i_s}^{k_s}$$

поставим в соответствие сочетание

$$a_{i_1} a_{i_2} \ldots a_{i_s} b_1 b_2 \ldots b_{k_1-1} b_{k_1+1} b_{k_1+2} \ldots$$
$$\ldots b_{k_1+k_2-1} b_{k_1+k_2+1} \ldots b_{k_1+\ldots+k_{s-1}+1} \ldots b_{k_1+\ldots+k_s-1}.$$

Как видно, объекты из множества $\{b_1, b_2, \ldots, b_{k-1}\}$ входят в данное сочетание массивами:

$$b_1 b_2 \ldots b_{k_1-1};\ b_{k_1+1} b_{k_1+2} \ldots b_{k_1+k_2-1};\ \ldots$$
$$\ldots;\ b_{k_1+\ldots+k_{s-1}+1} \ldots b_{k_1+\ldots+k_s-1}.$$

Число массивов равно s, и каждый из них содержит соответственно по $k_1 - 1$, $k_2 - 1$, ..., $k_s - 1$ объектов. Корректность такого построения вытекает из тождества

$$(k_1 - 1) + (k_2 - 1) + \ldots + (k_s - 1) + (s - 1) = k - 1.$$

Отсюда вытекает также, что число объектов в сочетании равно

$$s + (k_1 - 1) + (k_2 - 1) + \ldots + (k_s - 1) = k.$$

Таким образом, построенное сочетание содержит по одному представителю каждого сорта объектов, встречающихся в сочетании с повторениями, т. е. $a_{i_1}, \ldots, a_{i_s}$, а длины массивов объектов из множества $\{b_1, \ldots, b_{k-1}\}$ являются кодами кратностей вхождения объектов $a_{i_1}, \ldots, a_{i_s}$ в исходное сочетание с повторениями. При этом

кратность вхождения 1 кодируется 0,
» » 2 » 1,
. . . . . .
» » k_i » $k_i - 1$.

Если, например, взято сочетание с повторениями $a_2^3a_4^2$ $(n=6,\ k=5)$, то, согласно описанному алгоритму, этому сочетанию будет соответствовать сочетание $a_2a_4b_1b_2b_4$.

Покажем, что каждому сочетанию из множества $\{a_1, \ldots, a_n, b_1, \ldots, b_{k-1}\}$ по k объектов соответствует в вышеуказанном смысле единственное сочетание с повторениями (из которого оно получается).

В этом сочетании встречается s объектов множества $\{a_1, \ldots, a_n\}$ и $s \geqslant 1$: $a_{i_1}a_{i_2}\ldots a_{i_s}$. Тогда остальные $k-s$ объектов принадлежат множеству $\{b_1, \ldots, b_{k-1}\}$. Следовательно, $(k-1)-(k-s)=s-1$ объектов из множества $\{b_1, \ldots, b_{k-1}\}$ не входят в рассматриваемое сочетание. Пусть это будут объекты с номерами

$$k_1,\ k_1+k_2,\ \ldots,\ k_1+k_2+\ldots+k_{s-1} \qquad (k_1,\ k_2,\ \ldots,\ k_{s-1} \geqslant 1).$$

Этими объектами множество $\{b_1, \ldots, b_{k-1}\}$ разбивается на s кусков, некоторые из них могут быть пустыми (в случае, если соответствующее $k_i=1$). Длины полученных кусков равны соответственно

$$k_1-1,\ k_2-1,\ \ldots,\ k_s-1,$$

где $k_s=k-(k_1+\ldots+k_{s-1})$. Производя декодирование, получим сочетание с повторениями

$$a_{i_1}^{k_1}a_{i_2}^{k_2}\ldots a_{i_s}^{k_s}.$$

Из данного соответствия немедленно получаем, что $H_n^k = \binom{n+k-1}{k}$. Теорема доказана.

6. n-мерный куб размера k $(k \geqslant 2)$. Случай $k=2$ нами уже разобран. При $k \geqslant 2$, очевидно, $|E_k^n| = k^n$. Рассмотрим набор $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in E_k^n$. Этот набор можно характеризовать значениями $k_1, k_2, \ldots, k_r$, которые в нем встречаются, и кратностями $n_1, n_2, \ldots, n_r$ $(n_i > 0,\ i = 1, \ldots, r)$ вхождений этих значений в $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Очевидно, $n_1+n_2+\ldots+n_r=n$.

Специфику набора $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ можно задавать в виде следующей записи: $k_1^{n_1}, k_2^{n_2}, \ldots, k_r^{n_r}$. Совокупность наборов с данной спецификой $k_1^{n_1}, k_2^{n_2}, \ldots, k_r^{n_r}$ будем называть слоем. Подсчитаем число точек в данном слое.

Выбор позиций для значения k_1 осуществляется $\binom{n}{n_1}$ способами.

Далее, выбор позиций для значения k_2 осуществляется $\binom{n-n_1}{n_2}$ способами.

. . . . . .

Выбор позиций для значения k_r осуществляется $\binom{n-n_1-\ldots-n_{r-1}}{n_r}$ способами (т. е. однозначно).

Таким образом, интересующее нас число равно

$$\binom{n}{n_1}\binom{n-n_1}{n_2}\ldots\binom{n-n_1-\ldots-n_{r-1}}{n_r} = \frac{n!}{n_1!(n-n_1)!}\cdot\frac{(n-n_1)!}{n_2!\,(n-n_1-n_2)!}\cdot\ldots \cdot \ldots\cdot\frac{(n-n_1-\ldots-n_{r-1})!}{n_r!\,0!} = \frac{n!}{n_1!n_2!\ldots n_r!}.$$

Это число обозначается также через

$$\binom{n}{n_1, n_2, \ldots, n_r}.$$

Число слоев с заданными r значениями $k_1, \ldots, k_r$ равно числу решений уравнения

$$n_1 + n_2 + \ldots + n_r = n, \quad n_1, \ldots, n_r > 0. \tag{14}$$

И, наконец, число выборов из k каких-либо r значений равно $\binom{k}{r}$. Окончательно мы получаем

$$k^n = \sum_{r=1}^{k}\binom{k}{r}\sum_{\substack{(n_1,n_2,\ldots,n_r)\\ n_1+n_2+\ldots+n_r=n\\ (n_1,n_2,\ldots,n_r>0)}}\frac{n!}{n_1!n_2!\ldots n_r!}. \tag{15}$$

Эта формула при $k=2$ переходит в (9).

7. Разбиения множества E. Обозначим через $\Phi(n, k)$ число разбиений множества $E=\{a_1, \ldots, a_n\}$ на k $(n>0,\ 0<k\leqslant n)$ непустых частей, а через $\Phi(n)$ — число всех разбиений множества E $(n>0)$ на непустые части. Ино-

гда доопределяют эти числа для случая $k > n$, $k = 0$ и $n = 0$:

$$\Phi(n, k) = 0, \quad k > n,$$
$$\Phi(n, 0) = 0, \quad n > 0,$$
$$\Phi(0) = \Phi(0, 0) = 1.$$

Комбинаторные числа $\Phi(n, k)$ называются *числами Стирлинга 2-го рода*, а $\Phi(n)$ — *числами Белла*. Очевидно,

$$\Phi(n) = \sum_{k=0}^{n} \Phi(n, k). \tag{16}$$

Найдем сначала явную формулу для чисел $\Phi(n, k)$. Каждое разбиение $E = \mathscr{E}_1 \cup \mathscr{E}_2 \cup \ldots \cup \mathscr{E}_k$ на непустые подмножества можно характеризовать набором чисел $(l_1, l_2, \ldots, l_n)$, где

l_1 — число подмножеств разбиения мощности 1,
l_2 — число подмножеств разбиения мощности 2,
. . . . . .
l_n — число подмножеств разбиения мощности n.

Очевидно, что эти числа удовлетворяют тождеству

$$1 \cdot l_1 + 2l_2 + \ldots + nl_n = n. \tag{17}$$

Теорема 3.

$$\Phi(n, k) = \sum_{\substack{(l_1, l_2, \ldots, l_n) \\ l_1, l_2, \ldots, l_n \geqslant 0 \\ 1 \cdot l_1 + 2 \cdot l_2 + \ldots + nl_n = n \\ l_1 + l_2 + \ldots + l_n = k}} \frac{n!}{l_1! l_2! \ldots l_n! \, (1!)^{l_1} (2!)^{l_2} \ldots (n!)^{l_n}}.$$

Доказательство. Процесс построения всех разбиений множества E на k непустых частей, характеризуемых набором чисел $(l_1, l_2, \ldots, l_n)$, $l_1 + l_2 + \ldots + l_n = k$, можно представить следующим образом. Возьмем n упорядоченных ячеек и разобъем их на k подмножеств, характеризуемых данным набором чисел $(l_1, l_2, \ldots, l_n)$. Эти подмножества занумеруем числами $0, 1, \ldots, k-1$. Разместим в этих ячейках элементы $a_1, \ldots, a_n$. Очевидно, что разбиение ячеек на подмножества структуры $(l_1, l_2, \ldots, l_n)$ порождает разбиение элементов $a_1, \ldots, a_n$ на подмножества такой же структуры. Последнее задается набором $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, где α_i — номер подмножества разбиения ячеек, которому принадлежит элемент a_i. Производя различные размещения элементов $a_1, \ldots, a_n$

по ячейкам, мы получим все разбиения множества E на k непустых частей данной структуры $(l_1, l_2, \ldots, l_n)$. При этом два размещения определяют одно и то же разбиение множества E тогда и только тогда, когда для соответствующих им наборов $(\alpha_1', \alpha_2', \ldots, \alpha_n')$ и $(\alpha_1'', \alpha_2'', \ldots, \alpha_n'')$ выполнено условие: для любых i и j равенство $\alpha_i' = \alpha_j'$ эквивалентно равенству $\alpha_i'' = \alpha_j''$. Это означает, что два таких размещения переводятся друг в друга преобразованием, состоящим из перестановок элементов внутри одной компоненты разбиения и перестановок компонент разбиения, имеющих одинаковую мощность. Таким образом, среди $n!$ возможных размещений элементов каждое разбиение повторится ровно $l_1!l_2!\ldots l_n!\,(1!)^{l_1} \times (2!)^{l_2}\ldots(n!)^{l_n}$ раз. Теорема доказана.

Мы уже видели, что разбиения множества E связаны с наборами $(\alpha_1, \ldots, \alpha_n)$. Выберем r значений из множества $\{0, 1, \ldots, k-1\}$. Пусть это будут $k_1, k_2, \ldots, k_r$. Очевидно, таких выборов будет $\binom{k}{r}$. Возьмем произвольное разбиение E на r частей. Число разбиений равно $\Phi(n, r)$. Если нумеровать компоненты разбиений числами $k_1, k_2, \ldots, k_r$ ($r!$ способов), то мы получим всевозможные наборы длины n, содержащие ровно r значений $k_1, k_2, \ldots, k_r$. Очевидно, что каждый набор при этом будет построен ровно один раз. Поэтому

$$k^n = \sum_{r=1}^{k} \binom{k}{r} r!\,\Phi(n, r). \tag{18}$$

Если теперь сравнить соответствующие слагаемые в (15) и (18), то из рассуждения можно увидеть, что они выражают одно и то же число. Отсюда получаем еще одно явное выражение для $\Phi(n, r)$ $(n, r > 0)$:

$$\Phi(n, r) = \frac{1}{r!} \sum_{\substack{n_1+n_2+\ldots+n_r=n \\ n_1, n_2, \ldots, n_r > 0}} \frac{n!}{n_1!n_2!\ldots n_r!}. \tag{19}$$

Полученные формулы для $\Phi(n, k)$ практически не пригодны для вычисления $\Phi(n, k)$, так как они предполагают знание всех решений уравнения (14) или (17). Эффективные способы вычислений чисел Стирлинга 2-го рода и изучение их свойств связано с установлением ряда рекуррентных соотношений для $\Phi(n, k)$.

Возьмем произвольное разбиение E на k непустых подмножеств и выбросим одну из компонент (что возможно k способами). Оставшаяся часть множества E имеет мощность i $(k-1 \leqslant i \leqslant n-1)$ и разбита на $k-1$ частей. Таким образом, разбиению E на k непустых подмножеств соответствует k разбиений множеств мощности, меньшей или равной $n-1$, на $k-1$ часть. С другой стороны, если взять произвольное собственное подмножество в E и выбрать в нем любое разбиение на $k-1$ непустую часть, то оно может быть однозначно продолжено до разбиения множества E на k непустых частей.

Отсюда *)

$$k\Phi(n, k) = \sum_{i=k-1}^{n-1} \binom{n}{i} \Phi(i, k-1)$$

или

$$k\Phi(n, k) = \sum_{i=0}^{n-1} \binom{n}{i} \Phi(i, k-1). \tag{20}$$

Несколько видоизменим предыдущее рассуждение. В произвольном разбиении E на k непустых подмножеств, выбросим ту компоненту, которая содержит фиксированный элемент a_n $(a_n \in E)$. Тогда этому разбиению однозначным образом соответствует разбиение на $k-1$ непустых подмножеств некоторого множества из i элементов $(k-1 \leqslant i \leqslant n-1)$. Справедливо и обратное утверждение: любое разбиение на $k-1$ непустых частей произвольного подмножества из E, не содержащего a_n, однозначным образом продолжается до разбиения множества E на k непустых частей.

Поэтому

$$\Phi(n, k) = \sum_{i=k-1}^{n-1} \binom{n-1}{i} \Phi(i, k-1)$$

или

$$\Phi(n, k) = \sum_{i=0}^{n-1} \binom{n-1}{i} \Phi(i, k-1). \tag{21}$$

Почленно суммируя по k полученное тождество и учитывая (16), имеем

$$\Phi(n) = \sum_{i=0}^{n-1} \binom{n-1}{i} \Phi(i). \tag{22}$$

*) Эта формула верна и при $k = 1$, что вытекает из краевых условий (см. с. 184).

Заметим, что произвольное разбиение E на k непустых частей получается:

а) либо из разбиения множества $E\backslash a_n$ на $k-1$ непустую часть добавлением подмножества $\{a_n\}$;

б) либо из разбиения множества $E\backslash a_n$ на k непустых частей путем добавления к одной из них элемента a_n (k способов).

Отсюда получаем тождество

$$\Phi(n,\ k)=\Phi(n-1,\ k-1)+k\Phi(n-1,\ k). \qquad (23)$$

Оно позволяет построить таблицу для чисел $\Phi(n,\ k)$ (с линейной сложностью) и чисел Белла.

Таблица 3

n	k											
	0	1	2	3	4	5	6	7	8	9	...	$\Phi(n)$
0	1										...	1
1	0	1									...	1
2	0	1	1				0				...	2
3	0	1	3	1							...	5
4	0	1	7	6	1						...	15
5	0	1	15	25	10	1					...	52
6	0	1	31	90	65	15	1				...	203
7	0	1	63	301	350	140	21	1			...	877
8	0	1	127	966	1701	1050	266	28	1		...	4140
9	0	1	255	3025	7770	6951	2646	462	36	1	...	21147
...	...	...	...	...	...	...	...	...	...	...	...	...

Опираясь на рекуррентные соотношения для чисел $\Phi(n,\ k)$, докажем следующий факт.

Теорема 4. *Последовательность* $\{\Phi(n,\ k)\}$ *при фиксированном* n *и* $k=0,\ 1,\ \ldots,\ n$ *унимодальна: существует*

такое k_n, *что*

$$\Phi(n, 0) < \Phi(n, 1) < \Phi(n, 2) < \ldots < \Phi(n, k_n) \geqslant$$
$$\geqslant \Phi(n, k_n + 1) > \ldots > \Phi(n, n),$$

и $k_n = k_{n-1}$ *или* $k_n = k_{n-1} + 1$.

Доказательство. Ведем индукцией по n. При $n = 2$ утверждение очевидно (см. табл. 3).

Индуктивный переход от $n-1$ к n.

а) Пусть $2 \leqslant k \leqslant k_{n-1}$. Используя (23), имеем

$$\Phi(n, k) - \Phi(n, k-1) =$$
$$= (\Phi(n-1, k-1) - \Phi(n-1, k-2)) +$$
$$+ k(\Phi(n-1, k) - \Phi(n-1, k-1)) + \Phi(n-1, k-1) > 0.$$

б) Пусть $k_{n-1} + 2 \leqslant k \leqslant n$. Используя (21) и учитывая, что $k_i \leqslant k_{n-1}$ при $i < n-1$, имеем

$$\Phi(n, k) - \Phi(n, k-1) = \sum_{i=0}^{n-1} \binom{n-1}{i} \times$$
$$\times (\Phi(i, k-1) - \Phi(i, k-2)) < 0.$$

Сравним теперь величины $\Phi(n, k_{n-1})$ и $\Phi(n, k_{n-1}+1)$.

Если $\Phi(n, k_{n-1}) \geqslant \Phi(n, k_{n-1}+1)$, то полагаем $k_n = k_{n-1}$.

Если $\Phi(n, k_{n-1}) < \Phi(n, k_{n-1}+1)$, то полагаем $k_n = k_{n-1} + 1$.

Утверждение доказано.

Уже первое знакомство с комбинаторными объектами показывает, что мы сталкиваемся с общими задачами: такими, как задача о построении комбинаторных объектов и чисел (задача о перечислении), как задача о построении комбинаторных тождеств, как задача об изучении свойств комбинаторных чисел (например, наличие унимодальности) и т. п.

§ 3. Методы изучения комбинаторных объектов и чисел

В комбинаторном анализе существует целый ряд подходов для изучения комбинаторных объектов и чисел.

Теоретико-множественный подход. Он связан с вычислениями мощностей конечных подмножеств.

Пусть $A_1, \ldots, A_n$ — система подмножеств конечного множества A. Обозначим через H_i $(i=0, \ldots, n)$ совокупность всех элементов из A, которые содержатся ровно в i множествах системы, и через G_i $(i=0, 1, \ldots, n)$ — совокупность всех элементов из A, которые принадлежат не менее чем i множествам системы. Очевидно, что

$$A = H_0 \cup H_1 \cup \ldots \cup H_n$$

и

$$G_i = H_i \cup H_{i+1} \cup \ldots \cup H_n.$$

Возникает вопрос, как найти $|A_1 \cup \ldots \cup A_n|$, а также как находить $|A \setminus (A_1 \cup \ldots \cup A_n)|$, $|H_i|$ и $|G_i|$ $(i=0, 1, \ldots, n)$. Для решения этого вопроса необходима дополнительная информация, т. е. надо заранее знать мощности некоторых подмножеств.

Например, если известны мощности множеств

$$A_1^{\sigma_1} \cap A_2^{\sigma_2} \cap \ldots \cap A_n^{\sigma_n},$$

где

$$A_i^{\sigma_i} = \begin{cases} A_i & \text{при } \sigma_i = 1, \\ A \setminus A_i & \text{при } \sigma_i = 0, \end{cases}$$

то (по аналогии с совершенной д. н. ф.)

$$|A_1 \cup \ldots \cup A_n| = \sum_{\substack{(\sigma_1, \ldots, \sigma_n) \\ (\sigma_1, \ldots, \sigma_n) \neq (0, \ldots, 0)}} |A_1^{\sigma_1} \cap \ldots \cap A_n^{\sigma_n}|.$$

Оказывается, что решение поставленных вопросов возможно также, если известны мощности множеств

$$A_{i_1} \cap \ldots \cap A_{i_l}$$

для любых подмножеств чисел $\{i_1, \ldots, i_l\}$ $(l=1, 2, \ldots, n)$.

Теорема 5 (принцип включения-исключения).

$$|A \setminus (A_1 \cup \ldots \cup A_n)| = |A| - \sum_{i=1}^{n} |A_i| +$$

$$+ \sum_{1 \leqslant i_1 < i_2 \leqslant n} |A_{i_1} \cap A_{i_2}| - \ldots + (-1)^l \sum_{1 \leqslant i_1 < \ldots < i_l \leqslant n} |A_{i_1} \cap$$

$$\cap A_{i_2} \cap \ldots \cap A_{i_l}| + \ldots + (-1)^n |A_1 \cap A_2 \cap \ldots \cap A_n|.$$

Доказательство. Пусть $a \in A$ и a входит ровно в k множеств $(k=1, \ldots, n)$. Тогда $a \notin A \setminus (A_1 \cup \ldots \cup A_n)$.

С другой стороны, элемент a учитывается

в слагаемом $|A|$ 1 раз,

$\gg \sum_{i=1}^{n} |A_i|$ $\binom{k}{1}$ раз,

$\gg \sum_{1 \leqslant i_1 < i_2 \leqslant n} |A_{i_1} \cap A_{i_2}| \binom{k}{2}$ раз,

. . . . . .

Таким образом, его вклад в правую часть равен

$$\binom{k}{0} - \binom{k}{1} + \ldots + (-1)^k \binom{k}{k} = 0.$$

Если элемент a не входит в $(A_1 \cup \ldots \cup A_n)$, то он учитывается один раз в левой и в правой частях равенства. Теорема доказана.

Рассмотрим применение этой теоремы.

Пусть $A = P_2^n$ — множество всех булевых функций, зависящих от переменных $x_1, \ldots, x_n$; A_i — множество всех булевых функций из A, которые существенно не зависят от x_i. Очевидно,

$$|A_{i_1} \cap \ldots \cap A_{i_l}| = 2^{2^{n-l}}.$$

Подсчитаем число $\tilde{p}_n$ функций из A, существенно зависящих от всех переменных $x_1, \ldots, x_n$. Мы имеем

$$\tilde{p}_n = |A \setminus (A_1 \cup \ldots \cup A_n)|$$

и в силу теоремы 5

$$\tilde{p}_n = 2^{2^n} - \binom{n}{1} 2^{2^{n-1}} + \binom{n}{2} 2^{2^{n-2}} - \ldots + (-1)^n \binom{n}{n} 2.$$

Решение остальных вопросов, а также сам принцип включения-исключения могут быть получены из теоремы обращения, которая приводится ниже.

Алгебраический подход. Он основан на использовании вспомогательных просто получаемых комбинаторных тождеств для нахождения интересующих нас комбинаторных чисел.

Пусть имеются два семейства комбинаторных чисел $\{a_{n,k}\}$ и $\{b_{n,k}\}$, где $n = 0, 1, \ldots; k = 0, 1, \ldots, n$.

Теорема 6 (теорема обращения). *Если для любых* n *и* $k \leqslant n$

$$a_{n,k} = \sum_{i=0}^{n} \lambda_{n,k,i} b_{n,i}$$

и при $k \leqslant n$, $r \leqslant n$

$$\sum_{i=0}^{n} \mu_{n,k,i} \lambda_{n,i,r} = \begin{cases} 1 & \text{при } r = k, \\ 0 & \text{при } r \neq k, \end{cases} \tag{24}$$

то при $k \leqslant n$

$$b_{n,k} = \sum_{i=0}^{n} \mu_{n,k,i} a_{n,i}.$$

Доказательство. Используя (24), имеем

$$\sum_{i=0}^{n} \mu_{n,k,i} a_{n,i} = \sum_{i=0}^{n} \mu_{n,k,i} \sum_{r=0}^{n} \lambda_{n,i,r} b_{n,r} =$$

$$= \sum_{r=0}^{n} \left(\sum_{i=0}^{n} \mu_{n,k,i} \lambda_{n,i,r} \right) b_{n,r} = b_{n,k}.$$

Примеры. 1. Пусть $\lambda_{n,k,i} = \binom{k}{i}$, $\mu_{n,k,i} = (-1)^{k-i} \binom{k}{i}$. Тогда в силу (12) при $k \leqslant n$, $r \leqslant n$

$$\sum_{i=0}^{n} \mu_{n,k,i} \lambda_{n,i,r} = \sum_{i=0}^{n} (-1)^{k-i} \binom{k}{i} \binom{i}{r} = \sum_{i=r}^{k} (-1)^{k-i} \binom{k}{i} \binom{i}{r} =$$

$$= (-1)^{k-r} \sum_{i=0}^{k} (-1)^{i-r} \binom{k}{i} \binom{i}{r} = \begin{cases} 1 \text{ при } r = k, \\ 0 \text{ при } r \neq k. \end{cases}$$

Формулы обращения здесь имеют вид

$$a_{n,k} = \sum_{i=0}^{k} \binom{k}{i} b_{n,i}, \quad b_{n,k} = \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} a_{n,i}, \tag{25}$$

число n является параметром.

2. Пусть $\lambda_{n,k,i} = \binom{i}{k}$, $\mu_{n,k,i} = (-1)^{i-k} \binom{i}{k}$. Тогда, используя (12), имеем при $k \leqslant n$, $r \leqslant n$

$$\sum_{i=0}^{n} \mu_{n,k,i} \lambda_{n,i,r} = \sum_{i=0}^{n} (-1)^{i-k} \binom{i}{k} \binom{r}{i} =$$

$$= \sum_{i=0}^{r} (-1)^{i-k} \binom{r}{i} \binom{i}{k} = \begin{cases} 1 \text{ при } r = k, \\ 0 \text{ при } r \neq k. \end{cases}$$

Формулы обращения имеют вид

$$a_{n,k}=\sum_{i=k}^{n}\binom{i}{k}b_{n,i},\quad b_{n,k}=\sum_{i=k}^{n}(-1)^{i-k}\binom{i}{k}a_{n,i}. \tag{26}$$

Здесь n явно входит в пределы суммирования.

3. Пусть $\lambda_{n,k,i}=\binom{n-k}{i-k}$, $\mu_{n,k,i}=(-1)^{i-k}\binom{n-k}{i-k}$ при $i\geqslant k$ и $\lambda_{n,k,i}=\mu_{n,k,i}=0$ при $i<k$. Используя (13), имеем при $k\leqslant n$, $r\leqslant n$

$$\sum_{i=0}^{n}\mu_{n,k,i}\lambda_{n,i,r}=\sum_{i=0}^{n}(-1)^{i-k}\binom{n-k}{i-k}\binom{n-i}{r-i}=$$

$$=\sum_{i=k}^{r}(-1)^{i-k}\binom{n-i}{n-r}\binom{n-k}{n-i}=\begin{cases}1 & \text{при } r=k,\\ 0 & \text{при } r\neq k.\end{cases}$$

Соответствующие формулы обращения имеют вид

$$a_{n,k}=\sum_{i=k}^{n}\binom{n-k}{i-k}b_{n,i},\quad b_{n,k}=\sum_{i=k}^{n}(-1)^{i-k}\binom{n-k}{i-k}a_{n,i}. \tag{27}$$

Применения формул обращения. 1. Подсчет числа $\tilde{p}_n$ — булевых функций, зависящих существенно от данных n переменных. Очевидно,

$$2^{2^n}=\binom{n}{n}\tilde{p}_n+\binom{n}{n-1}\tilde{p}_{n-1}+\ldots+\binom{n}{0}\tilde{p}_0,\quad n=1,2,\ldots$$

Применяя формулу обращения (25), получаем

$$\tilde{p}_n=\binom{n}{n}2^{2^n}-\binom{n}{n-1}2^{2^{n-1}}+\ldots+(-1)^n\binom{n}{0}2^{2^0}. \tag{28}$$

2. Получение явной формулы для чисел Стирлинга 2-го рода $\Phi(n,k)$. Формула (18) имеет вид

$$k^n=\binom{k}{k}\Phi(n,k)k!+$$

$$+\binom{k}{k-1}\Phi(n,k-1)(k-1)!+\ldots+\binom{k}{0}\Phi(n,0)0!$$

Используя формулу обращения (25), получаем

$$\Phi(n,k)\,k!=\binom{k}{k}k^n-\binom{k}{k-1}(k-1)^n+\ldots+(-1)^k\binom{k}{0}\cdot 0^n$$

или

$$\Phi(n,k)=\frac{\binom{k}{k}k^n-\binom{k}{k-1}(k-1)^n+\ldots+(-1)^k\binom{k}{0}\cdot 0^n}{k!}.$$

Последнее выражение может быть преобразовано к виду

$$\Phi(n, k) = \frac{k^n}{k!} - \frac{(k-1)^n}{1!\,(k-1)!} + \frac{(k-2)^n}{2!\,(k-2)!} - \ldots$$

$$\ldots + (-1)^{k-1}\frac{1^n}{(k-1)!\,1!}. \quad (29)$$

Из выражений для $\Phi(n, k)$ можно получить явное выражение для чисел Белла $\Phi(n)$:

$$\Phi(n) = \sum_{k=0}^{n}\Phi(n,k) = \sum_{k=0}^{n}\sum_{i=0}^{k}(-1)^i\frac{\binom{k}{i}}{k!}(k-i)^n =$$

$$= \sum_{k=0}^{n}\sum_{i=0}^{k}(-1)^i\frac{(k-i)^n}{i!\,(k-i)!}.$$

Пусть $j = k - i$ и будем суммировать по параметрам i и j

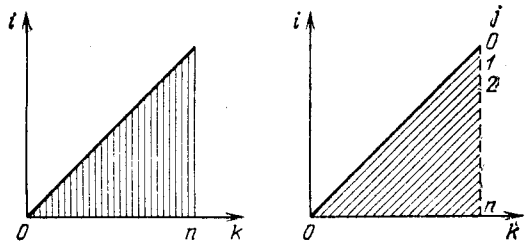


Рис. 4

($k = i + j$). Пределы суммирования легко усматриваются из рис. 4:

$$\Phi(n) = \sum_{j=0}^{n}\sum_{i=0}^{n-j}(-1)^i\frac{j^n}{i!j!} = \sum_{j=0}^{n}\frac{j^n}{j!}\sum_{i=0}^{n-j}(-1)^i\frac{1}{i!} =$$

$$= \frac{n^n}{n!} + \left(1 - \frac{1}{1!}\right)\frac{(n-1)^n}{(n-1)!} + \left(1 - \frac{1}{1!} + \frac{1}{2!}\right)\frac{(n-2)^n}{(n-2)!} + \ldots$$

$$\ldots + \left(1 - \frac{1}{1!} + \frac{1}{2!} - \ldots + (-1)^{n-1}\frac{1}{(n-1)!}\right)$$

или

$$\Phi(n) = \frac{n^n}{n!} + \left(1 - \frac{1}{1!} + \frac{1}{2!}\right)\frac{(n-2)^n}{(n-2)!} + \ldots$$

$$\ldots + \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \ldots + (-1)^{n-1}\frac{1}{(n-1)!}\right) \quad (30)$$

(здесь коэффициент при $j^n/j!$ — кусок ряда e^{-1}).

3. Вывод формулы включения-исключения и ее обобщений из формул обращения.

Обозначим $|A| = g_0, \sum_i |A_i| = g_1, \ldots, \sum_{1 \leqslant i_1 < \ldots < i_k \leqslant n} |A_{i_1} \cap \cap \ldots \cap A_{i_k}| = g_k, \ \ldots, \ g_n = |A_1 \cap \ldots \cap A_n|$ и $|H_i| = h_i$ $(i = 0, \ldots, n)$.

Напомним, что

$$|A| = \sum_{i=0}^{n} |H_i| = \sum_{i=0}^{n} h_i$$

и каждый элемент $a \in H_i$ входит в $\binom{i}{k}$ множеств вида $A_{i_1} \cap \ldots \cap A_{i_k}$ $(1 \leqslant i_1 < \ldots < i_k \leqslant n)$. Поэтому

$$g_0 = |A| = \binom{0}{0} h_0 + \binom{1}{0} h_1 + \ldots + \binom{n}{0} h_n,$$

$$g_1 = \sum_i |A_i| = \binom{1}{1} h_1 + \ldots + \binom{n}{1} h_n,$$

. . . . . .

$$g_k = \sum_{1 \leqslant i_1 < \ldots < i_k \leqslant n} |A_{i_1} \cap \ldots \cap A_{i_k}| = \binom{k}{k} h_k + \ldots + \binom{n}{k} h_n,$$

. . . . . .

$$g_n = |A_1 \cap \ldots \cap A_n| = \binom{n}{k} h_n.$$

Формулы обращения (26) дают

$$h_k = \sum_{i=k}^{n} (-1)^{i-k} \binom{i}{k} g_i$$

и при $k = 0$ имеем формулу включения-исключения:

$$h_0 = \sum_{i=0}^{n} (-1)^i g_i.$$

4. Подсчет числа неизбыточных (тупиковых) покрытий множества $A = \{a_1, \ldots, a_n\}$ его подмножествами $A_1, \ldots, A_k$:

$$A = A_1 \cup \ldots \cup A_k.$$

Покрытие называется *неизбыточным* (*тупиковым*), если при выбрасывании любого из его подмножеств A_i система подмножеств $A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n$ не образует покрытия A, т. е.

$$A \neq A_1 \cup \ldots \cup A_{i-1} \cup A_{i+1} \cup \ldots \cup A_n.$$

Очевидно, что в тупиковом покрытии все подмножества попарно различны.

В дальнейшем мы будем рассматривать более широкий объект: упорядоченные покрытия (т. е. подмножества занумерованы и допускается повторение подмножеств).

Каждому такому покрытию соответствует булева матрица $\|\alpha_{ij}\|$ порядка $k \times n$: в ней столбцы соответствуют элементам множества $A = \{a_1, \ldots, a_n\}$, а строки — подмножествам $A_1, \ldots, A_k$,

$$\alpha_{ij} = \begin{cases} 1, & \text{если } a_j \in A_i, \\ 0, & \text{если } a_j \notin A_i. \end{cases}$$

Поскольку матрица $\|\alpha_{ij}\|$ соответствует покрытию множества A, то она не содержит нулевых столбцов. При фиксированных k и n число таких матриц равно $(2^k - 1)^n$.

Обозначим через $A(i_1, \ldots, i_s)$ подмножество из указанных матриц таких, что в каждой из них, если удалить любую из строк $i_1, \ldots, i_s$, то получим матрицу, соответствующую покрытию множества A; через $B(i_1, \ldots, i_s)$ обозначим подмножество множества $A(i_1, \ldots, i_s)$ таких матриц, в которых (в указанном смысле) невозможно удаление любой строки с номером, отличным от $i_1, \ldots, i_s$.

В силу определения матрицы из $A(i_1, \ldots, i_s)$ не содержат столбцов, имеющих ровно одну единицу в строках $i_1, \ldots, i_s$. Поэтому

$$|A(i_1, \ldots, i_s)| = (2^k - s - 1)^n.$$

Из формулы видно, что $|A(i_1, \ldots, i_s)|$ зависит от s, но не зависит от выбора строк. Эту величину обозначим далее через $a^n_{k,s}$. Аналогично $|B(i_1, \ldots, i_s)|$ также не зависит от выбора строк, поэтому положим $|B(i_1, \ldots, i_s)| = b^n_{k,s}$.

Пусть $\|\alpha_{ij}\| \in A(i_1, \ldots, i_s)$. Рассмотрим в ней все номера строк, которые допускают удаление (сюда, в частности, войдут и номера $i_1, \ldots, i_s$). Пусть $i_1, \ldots, i_s, i_{s+1}, \ldots, i_{s+r}$ — их номера и $s + r = m$. Тогда $\|\alpha_{ij}\| \in B(i_1, \ldots, i_s, i_{s+1}, \ldots, i_m)$.

Отсюда

$$a^n_{k,s} = \sum_{r=0}^{k-s} \binom{k-s}{r} b^n_{k,s+r} = \sum_{m=s}^{k} \binom{k-s}{m-s} b^n_{k,m}.$$

В силу формулы обращения (27)

$$b^n_{k,s} = \sum_{m=s}^{k} (-1)^{m-s} \binom{k-s}{m-s} a^n_{k,m}$$

или

$$b^n_{k,s} = \sum_{m=s}^{k} (-1)^{m-s} \binom{k-s}{m-s} (2^k - m - 1)^n.$$

В частности, при $s = 0$

$$b^n_{k,0} = \sum_{m=0}^{k} (-1)^m \binom{k}{m} (2^k - m - 1)^n.$$

Отсюда получаем для числа $\tilde{b}^n_{k,0}$ (неупорядоченных) тупиковых покрытий длины k выражение

$$\tilde{b}^n_{k,0} = \frac{b^n_{k,0}}{k!} = \frac{1}{k!} \sum_{m=0}^{k} (-1)^m \binom{k}{m} (2^k - m - 1)^n$$

и для числа $\tilde{b}(n)$ всех тупиковых покрытий ($k \leqslant n$) — выражение *)

$$\tilde{b}(n) = \sum_{k=0}^{n} \frac{1}{k!} \sum_{m=0}^{k} (-1)^m \binom{k}{m} (2^k - m - 1)^n.$$

Алгебраические соображения часто используются для построения рекуррентных формул. Эту ситуацию проиллюстрируем на примере получения рекуррентной формулы для чисел

$$S_m(n) = 1^m + 2^m + \ldots + n^m.$$

Возьмем формулу бинома Ньютона

$$(n+1)^m = n^m + \binom{m}{1} n^{m-1} + \binom{m}{2} n^{m-2} + \ldots + \binom{m}{m-1} n + \binom{m}{m}.$$

---

*) Это число совадает с числом тупиковых проверяющих тестов для универсальной таблицы $2^n \times n$.

С ее помощью имеем систему равенств

$$(n+1)^m = n^m + \binom{m}{1} n^{m-1} + \binom{m}{2} n^{m-2} + \ldots + \binom{m}{m-1} n + \binom{m}{m},$$

$$[(n-1)+1]^m = (n-1)^m + \binom{m}{1}(n-1)^{m-1} + \binom{m}{2}(n-1)^{m-2} + \ldots$$

$$\ldots + \binom{m}{m-1}(n-1) + \binom{m}{m},$$

. . . . . .

$$(2+1)^m = 2^m + \binom{m}{1} 2^{m-1} + \binom{m}{2} 2^{m-2} + \ldots + \binom{m}{m-1} 2 + \binom{m}{m},$$

$$(1+1)^m = 1^m + \binom{m}{1} 1^{m-1} + \binom{m}{2} 1^{m-2} + \ldots + \binom{m}{m-1} 1 + \binom{m}{m}.$$

Складывая их почленно, получаем рекуррентную формулу

$$(n+1)^m = 1 + \binom{m}{1} S_{m-1}(n) + \binom{m}{2} S_{m-2}(n) + \ldots$$

$$\ldots + \binom{m}{m-1} S_1(n) + \binom{m}{m} S_0(n),$$

где $S_0(n) = n$, $S_1(n) = \frac{n(n+1)}{2}$.

Отсюда можно постепенно найти $S_m(n)$. Например,

$$(n+1)^3 = 1 + \binom{3}{1} S_2(n) + \binom{3}{2} S_1(n) + \binom{3}{3} S_0(n),$$

$$3S_2(n) = (n+1)^3 - 1 - 3\frac{n(n+1)}{2} - n = \frac{n(n+1)(2n+1)}{2}$$

или

$$S_2(n) = \frac{n(n+1)(2n+1)}{6}.$$

Метод производящих функций. Он используется для перечисления комбинаторных чисел и установления комбинаторных тождеств.

Исходным пунктом являются последовательность $\{a_i\}$ комбинаторных чисел и последовательность функций $\{\varphi_i(x)\}$ $(i = 0, 1, \ldots)$.

Рассматриваем далее ряд

$$\sum_{i=0}^{\infty} a_i \varphi_i(x),$$

который, в случае, когда последовательность $\{a_i\}$ конечна, т. е. $0 \leqslant i \leqslant n$, будет многочленом. При определенных ограничениях данный ряд будет сходящимся и тогда он в некоторой области будет задавать функцию $F(x)$:

$$F(x) = \sum_{i=0}^{\infty} a_i \varphi_i(x).$$

Эта функция называется *производящей функцией.*

Рассмотрим ряд примеров, относящихся к типичным случаям.

1. $a_i = \binom{n}{i}$ $(i = 0, 1, \ldots, n)$, $\varphi_i(x) = x^i$. В этом случае мы уже имели

$$(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i.$$

В качестве производящей функции здесь будет бином Ньютона $(1+x)^n$. С помощью производящей функции установим тождество

$$\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k}^2.$$

Для этого возьмем тождество

$$(1+x)^{2n} = (1+x)^n (1+x)^n.$$

Оно эквивалентно тождеству

$$\sum_{i=0}^{2n} \binom{2n}{i} x^i = \left(\sum_{k=0}^{n} \binom{n}{k} x^k \right) \sum_{m=0}^{n} \binom{n}{m} x^m \Big).$$

Сравнивая коэффициенты при x^n, получим

$$\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k} \binom{n}{n-k} = \sum_{k=0}^{n} \binom{n}{k}^2.$$

2. Рассмотрим пример на применение метода производящих функций, когда функция определяется степенным рядом.

Как известно, последовательность чисел f_n, называемых *числами Фибоначчи*, задается рекуррентными соотношениями:

$$f_n = f_{n-1} + f_{n-2} \quad \text{и} \quad f_0 = f_1 = 1.$$

Возьмем $\varphi_n(x) = x^n$ $(n = 0, 1, \ldots)$. С этой последовательностью связан ряд

$$\sum_{n=0}^{\infty} f_n x^n,$$

который в силу $f_n \leqslant 2^n$ (поскольку $f_n \leqslant 2f_{n-1}$) сходится при $|x| < 1/2$ и определяет производящую функцию $F(x)$:

$$F(x) = \sum_{n=0}^{\infty} f_n x^n.$$

Так как

$$xF(x) = \sum_{n=1}^{\infty} f_{n-1} x^n$$

и

$$x^2 F(x) = \sum_{n=2}^{\infty} f_{n-2} x^n,$$

то

$$xF(x) + x^2 F(x) = \sum_{n=2}^{\infty} (f_{n-1} + f_{n-2}) x^n + x = F(x) - 1$$

или

$$(1 - x - x^2) F(x) = 1.$$

Отсюда находим явный вид производящей функции $F(x)$:

$$F(x) = \frac{1}{1 - x - x^2}.$$

Решая уравнение

$$x^2 + x - 1 = 0,$$

находим его корни

$$x_{1,2} = \frac{-1 \pm \sqrt{5}}{2}.$$

Найдем разложение $F(x)$ на элементарные дроби:

$$\frac{1}{1 - x - x^2} = \frac{a}{x_1 - x} + \frac{b}{x_2 - x}.$$

Имеем

$$ax_2 + bx_1 - (a + b)x \equiv -1.$$

Это справедливо, если

$$a = -b \text{ и } b = -\frac{1}{x_1 - x_2} = -\frac{1}{\sqrt{5}}.$$

Далее, воспользовавшись формулой для суммы убывающей геометрической прогрессии $\left(\text{при } \left|\frac{x}{x_1}\right|<1,\ \left|\frac{x}{x_2}\right|<1\right)^*$), получим

$$\frac{1}{1-x-x^2}=\frac{1}{\sqrt{5}}\left(\frac{1}{x_1-x}-\frac{1}{x_2-x}\right)=$$

$$=\frac{1}{\sqrt{5}}\left(\frac{1}{x_1}\frac{1}{1-\frac{x}{x_1}}-\frac{1}{x_2}\frac{1}{1-\frac{x}{x_2}}\right)=$$

$$=\frac{1}{\sqrt{5}}\left(\frac{1}{x_1}\sum_{n=0}^{\infty}\left(\frac{x}{x_1}\right)^n-\frac{1}{x_2}\sum_{n=0}^{\infty}\left(\frac{x}{x_2}\right)^n\right)=$$

$$=\frac{1}{\sqrt{5}}\sum_{n=0}^{\infty}\frac{x_2^{n+1}-x_1^{n+1}}{(x_1x_2)^{n+1}}x^n=\frac{1}{\sqrt{5}}\sum_{n=0}^{\infty}\left[(-x_2)^{n+1}-(-x_1)^{n+1}\right]x^n,$$

откуда

$$f_n=\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1}-\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right],$$

что дает явное выражение для чисел Фибоначчи.

3. В случае, если рост комбинаторных чисел достаточно велик, используются так называемые экспоненциальные производящие функции. Пусть $a_n=\Phi(n)$ и $\varphi_n(z)=z^n/n!$ Рассмотрим ряд

$$\sum_{n=0}^{\infty}\frac{\Phi(n)}{n!}z^n.$$

Теорема 7 (Белл). *Имеет место следующее тождество*

$$e^{e^z-1}=\sum_{n=0}^{\infty}\frac{\Phi(\ )}{n!}z^n.$$

Доказательство. Очевидно, что

$$e^{e^z-1}=e^{z+z^2/2!+\ldots}=e^z\cdot e^{z^2/2!}\ldots e^{z^n/n!}\ldots,$$

*) Соответствующие ряды также сходятся в окрестности $x=0$.

где

$$e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \ldots + \frac{z^{l_1}}{l_1!} + \ldots,$$

$$e^{z^2/2!} = 1 + \frac{z^2}{2!} + \frac{z^4}{2!\,(2!)^2} + \ldots + \frac{z^{2l_2}}{l_2!(2!)^{l_2}} + \ldots,$$

$$e^{z^3/3!} = 1 + \frac{z^3}{3!} + \frac{z^6}{2!\,(3!)^2} + \ldots + \frac{z^{3l_3}}{l_3!\,(3!)^{l_3}} + \ldots,$$

. . . . . .

Перемножая эти ряды и собирая члены с z^n, находим, что коэффициент при z^n равен

$$\sum_{\substack{(l_1, l_2, \ldots, l_n) \\ l_1+2l_2+\ldots+nl_n=n}} \frac{1}{(l_1!)\,(1!)^{l_1}\,(l_2!)\,(2!)^{l_2}\ldots(l_n!)\,(n!)^{l_n}}$$

или, что то же самое,

$$\frac{1}{n!} \sum_{\substack{(l_1, l_2, \ldots, l_n) \\ l_1+2l_2+\ldots+nl_n=n}} \frac{n!}{l_1!\,l_2!\ldots l_n!\,(1!)^{l_1}\,(2!)^{l_2}\ldots(n!)^{l_n}} = \frac{1}{n!}\,\Phi(n)$$

(последнее в силу теоремы 3). Теорема доказана.

Данное тождество может быть обобщено следующим образом:

$$e^{x(e^z-1)} = \sum_{n=0}^{\infty} \frac{l_n(x)\,z^n}{n!},$$

где

$$l_n(x) = \sum_{k=0}^{n} \Phi(n, k)\,x^k.$$

Доказательство аналогично. Собирая в соответствующем произведении рядов члены с $z^n x^k$ и замечая, что $l_1 + \ldots + l_n = k$, находим, что коэффициент при $z^n x^k$ равен

$$\frac{1}{n!} \sum_{\substack{(l_1, l_2, \ldots, l_n) \\ l_1+l_2+\ldots+l_n=k \\ l_1+2l_2+\ldots+nl_n=n}} \frac{n!}{l_1!\,l_2!\ldots l_n!\,(1!)^{l_1}\,(2!)^{l_2}\ldots(n!)^{l_n}} = \frac{\Phi(n, k)}{n!},$$

Так как $l_n(1)=\Phi(n)$, то как следствие имеем

$$e^{e^z-1}=\sum_{n=0}^{\infty}\frac{\Phi(n)\,z^n}{n!}\quad\text{(тождество Белла).}$$

Выполняя очевидные преобразования и пользуясь (29), получаем

$$e^{-x}\sum_{k=0}^{\infty}\frac{k^n x^k}{k!}=\left(1-\frac{x}{1!}+\frac{x^2}{2!}-\ldots\right)\sum_{k=0}^{\infty}\frac{k^n x^k}{k!}=$$

$$=\sum_{k=0}^{\infty}\left(\frac{k^n}{k!}-\frac{(k-1)^n}{1!\,(k-1)!}+\frac{(k-2)^n}{2!\,(k-2)!}-\ldots\right)x^k=$$

$$=\sum_{k=0}^{\infty}\Phi(n,k)\,x^k=\sum_{k=0}^{n}\Phi(n,k)\,x^k=l_n(x).$$

Полагая здесь $x=1$, получаем

$$\Phi(n)=\frac{1}{e}\sum_{k=0}^{\infty}\frac{k^n}{k!}\quad\text{(тождество Добинского).}$$

§ 4. Оценки и асимптотики для комбинаторных чисел

Проблематика этого параграфа использует ряд понятий из математического анализа [8, 14, 26], которые позволяют сравнивать поведение функций в окрестности некоторой точки. Напомним некоторые из них.

Пусть $f(x)$ и $g(x)$ — действительные функции, определенные на некотором подмножестве $\mathscr{E}$ действительных чисел, и a — некоторое число $(-\infty\leqslant a\leqslant+\infty)$. Число a является *предельной точкой* для $\mathscr{E}$, если в любой окрестности a содержится точка из множества $\mathscr{E}$, отличная от a.

Положим $f(x)=O(g(x))$ при $x\to a$, если существует такая константа C, $C>0$, что в некоторой окрестности a, исключая, быть может, a,

$$|f(x)|\leqslant C|g(x)|.$$

Примеры. 1. Пусть $f(x)=x$, $g(x)=x^2$ и $a=+\infty$. Тогда $x=O(x^2)$ при $x\to+\infty$ (здесь $x\leqslant x^2$, если $x\geqslant 1$).

2. Пусть $f(x)=\ln(1+x)-\left(x-\frac{x^2}{2}\right)$, $g(x)=x^3$ и $a=0$.

Тогда

$$\ln(1+x)-\left(x-\frac{x^2}{2}\right)=O(x^3) \quad \text{при} \quad x\to 0.$$

Данное определение эквивалентно следующему: $f(x)=O(g(x))$ при $x\to a$, если существует функция $\varphi(x)$ такая, что в некоторой окрестности a, исключая, быть может, a,

$$f(x)=\varphi(x)g(x) \quad \text{и} \quad |\varphi(x)|\leqslant C.$$

В случае, если в некоторой окрестности a (исключая, быть может, a) $g(x)\neq 0$, то определение допускает видоизменение: $f(x)=O(g(x))$ при $x\to a$, если существует такая константа C, $C>0$, что в некоторой окрестности a, исключая, быть может, a,

$$\left|\frac{f(x)}{g(x)}\right|\leqslant C.$$

Положим $f(x)\asymp g(x)$ при $x\to a$, если при $x\to a$ имеют место одновременно $f(x)=O(g(x))$ и $g(x)=O(f(x))$. В этом случае функции $f(x)$ и $g(x)$ имеют один и тот же порядок (при $x\to a$), а их знаки в окрестности a друг с другом, вообще говоря, никак не связаны.

Положим $f(x)=o(g(x))$ при $x\to a$, если существует функция $\varphi(x)$ такая, что в некоторой окрестности a, исключая, быть может, a,

$$f(x)=\varphi(x)g(x) \quad \text{и} \quad \varphi(x)\to 0 \quad \text{при} \quad x\to a.$$

Примеры. 1. Пусть $f(x)=x$, $g(x)=x^2$ и $a=+\infty$. Тогда $x=o(x^2)$ при $x\to+\infty$. Здесь $\varphi(x)=1/x$ и $\varphi(x)\to 0$ при $x\to+\infty$.

2. Пусть $f(x)=x^n$, $g(x)=x^{n-\varepsilon}$ ($\varepsilon>0$ — малый параметр) и $a=0$. Тогда $f(x)=o(g(x))$ при $x\to 0$.

Очевидно, что если в некоторой окрестности a (исключая, быть может, a) $g(x)\neq 0$, то условие $f(x)=o(g(x))$ при $x\to a$ эквивалентно условию

$$\frac{f(x)}{g(x)}\to 0 \quad \text{при} \quad x\to a.$$

Для отношений «O» и «o» имеют место следующие свойства.

1. Свойство транзитивности. Если $f(x)=O(g(x))$ и $g(x)=O(h(x))$ при $x\to a$, то $f(x)=O(h(x))$ при $x\to a$.

Если $f(x)=o(g(x))$ и $g(x)=o(h(x))$ при $x\to a$, то $f(x)=o(h(x))$ при $x\to a$.

2. Если $f_1(x)=O(g(x))$ и $f_2(x)=O(g(x))$ при $x\to a$, то $f_1(x)+f_2(x)=O(g(x))$ при $x\to a$.

Если $f_1(x)=o(g(x))$ и $f_2(x)=o(g(x))$ при $x\to a$, то $f_1(x)+f_2(x)=o(g(x))$ при $x\to a$.

3. Пусть $\psi(x)$ — произвольная функция, определенная в окрестности a. Тогда

из условия $f(x)=O(g(x))$ при $x\to a$ вытекает, что $f(x)\psi(x)=O(g(x)\psi(x))$ при $x\to a$;

из условия $f(x)=o(g(x))$ при $x\to a$ вытекает, что $f(x)\psi(x)=o(g(x)\psi(x))$ при $x\to a$.

4. Из условия $f(x)=o(g(x))$ при $x\to a$ следует, что $f(x)=O(g(x))$ при $x\to a$.

Положим $f(x)\sim g(x)$ при $x\to a$, если существует функция $\varphi(x)$ такая, что в некоторой окрестности a, исключая, быть может, a,

$$f(x)=\varphi(x)g(x) \quad \text{и} \quad \varphi(x)\to 1 \quad \text{при} \quad x\to a.$$

В этом случае говорят, что $f(x)$ *эквивалентна* (*асимптотически равна*) функции $g(x)$ при $x\to a$.

Примеры: $\sin x\sim x$ при $x\to 0$, $e^x\sim 1-x$ при $x\to 0$.

В случае, если $g(x)\neq 0$ (или $f(x)\neq 0$) в некоторой окрестности a (исключая, быть может, a), то условие $f(x)\sim g(x)$ при $x\to a$ эквивалентно условию

$$\frac{f(x)}{g(x)}\to 1 \quad \text{при} \quad x\to a.$$

Легко видеть, что условие $f(x)\sim g(x)$ при $x\to a$ эквивалентно условию

$$f(x)=g(x)+o(g(x)) \quad \text{при} \quad x\to a.$$

Замечание. Из условия $f(x)=g(x)+o(1)$ при $x\to a$ следует, что $e^{f(x)}\sim e^{g(x)}$ при $x\to a$.

В то же время, если $f(x)\sim g(x)$ при $x\to a$, то отсюда не следует, что $e^{f(x)}\sim e^{g(x)}$ при $x\to a$.

Пример. Пусть $\mathfrak{N}$ — множество натуральных чисел и $a=+\infty$. Пусть $f(n)=n$ и $g(n)=n+c$ $(c\neq 0)$. Ясно, что $f(n)\sim g(n)$ при $n\to\infty$, а

$$\frac{e^n}{e^{n+c}}=e^{-c}\not\to 1 \quad \text{при} \quad n\to+\infty.$$

Введем отношения $f(x)\lesssim g(x)$ при $x\to a$ и $f(x)\lesssim g(x)$ при $x\to a$, которые тесно связаны с предыдущими и являются естественными обобщениями отношения $\leqslant$.

Положим $f(x) \lesssim g(x)$ при $x \to a$, если существуют такие положительные константы C_1 и C_2 ($C_1 \geqslant 1$, $C_2 \leqslant 1$) и окрестность точки a, в которой (кроме, быть может, самой точки a)

$$f(x) \leqslant C(x) g(x),$$

где

$$C(x) = \begin{cases} C_1 & \text{при } g(x) > 0, \\ 1 & \text{при } g(x) = 0, \\ C_2 & \text{при } g(x) < 0. \end{cases}$$

Из определения следует, что график функции $f(x)$ лежит не выше графика функции $g(x)$, поднятого вверх в $C(x)$ раз.

Если при $x \to a$ одновременно $f(x) \lesssim g(x)$ и $g(x) \lesssim f(x)$, то пишем

$$f(x) \asymp g(x) \quad \text{при} \quad x \to a.$$

Покажем, что в этом случае функции $f(x)$ и $g(x)$ имеют один и тот же порядок и одинаковый знак. Действительно, при этом условии в некоторой окрестности a (исключая, быть может, a) одновременно выполняются неравенства

$$f(x) \leqslant C'(x) g(x) \quad \text{и} \quad g(x) \leqslant C''(x) f(x).$$

Из них получаем

$$f(x) \leqslant C'(x) g(x) \leqslant C'(x) C''(x) f(x)$$

и

$$g(x) \leqslant C''(x) f(x) \leqslant C''(x) C'(x) g(x).$$

Отсюда следует, что в указанной окрестности:

1. Функции $f(x)$ и $g(x)$ обращаются в 0 в одних и тех же точках.

2. В ненулевых точках функции $f(x)$ и $g(x)$ имеют один знак и в них

$$\widetilde{C}' \leqslant \left| \frac{f(x)}{g(x)} \right| \leqslant \widetilde{C}'',$$

т. е. $f(x) = O(g(x))$ и $g(x) = O(f(x))$ при $x \to a$. Следовательно, при $x \to a$ условие $f(x) \asymp g(x)$ влечет условие $f(x) \eqsim g(x)$.

Положим $f(x) \prec g(x)$ при $x \to a$, если для любого $\varepsilon > 0$ существует окрестность точки a, в которой (кроме,

быть может, точки a)

$$f(x) \leqslant C_\varepsilon(x)\, g(x),$$

где

$$C_\varepsilon(x) = \begin{cases} 1+\varepsilon & \text{при} \quad g(x) > 0, \\ 1 & \text{при} \quad g(x) = 0, \\ 1-\varepsilon & \text{при} \quad g(x) < 0. \end{cases}$$

Ясно, что из $f(x) \preccurlyeq g(x)$ при $x \to a$ вытекает, что $f(x) \lesssim g(x)$ при $x \to a$. Поэтому все сказанное выше об отношении $\lesssim$ справедливо и для отношения $\preccurlyeq$. В частности, в случае, когда при $x \to a$ одновременно $f(x) \preccurlyeq g(x)$ и $g(x) \preccurlyeq f(x)$, функции $f(x)$ и $g(x)$ обращаются в 0 в одних и тех же точках и в ненулевых точках выполняются неравенства (при $\varepsilon < 1$):

$$1-\varepsilon \leqslant \frac{f(x)}{g(x)} \leqslant \frac{1}{1-\varepsilon},$$

т. е. $f(x) \sim g(x)$ при $x \to a$.

Теперь перейдем к рассмотрению асимптотики (предельных теорем) для ряда комбинаторных чисел.

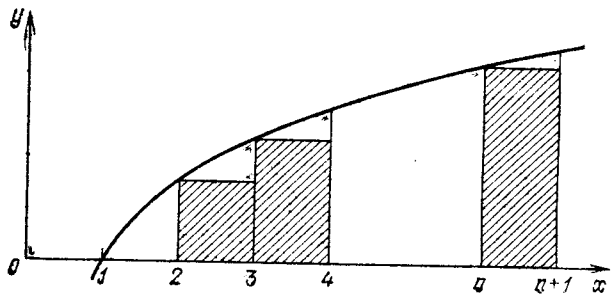


Рис. 5

Асимптотика $\ln n!$ Порядок $n!$

Теорема 8.

$$\ln n! \sim \left(n + \frac{1}{2}\right) \ln n.$$

Доказательство. Возьмем выражение

$$\ln n! = \sum_{i=1}^{n} \ln i.$$

Рассмотрим график функции $y = \ln x$ (см. рис. 5). Сумму $\sum_{i=1}^{n} \ln i$ можно рассматривать как площадь прямоугольников, вписанных в кривую $y = \ln x$. К этой величине добавим площадь треугольников (см. рис. 6), т. е. сумму

$$\sum_{i=1}^{n} \frac{1}{2} (\ln (i+1) - \ln i) = \frac{1}{2} \ln (n+1).$$

Получим

$$\ln n! + \frac{1}{2} \ln (n+1) < \int_1^{n+1} \ln x \, dx = x \ln x - x \Big|_1^{n+1} =$$

$$= (n+1) \ln (n+1) - n.$$

Отсюда

$$\ln n! < \left(n + \frac{1}{2}\right) \ln (n+1) - n.$$

С другой стороны, рассмотрим фигуру, образованную отрезками касательных в точках $x = 1, 2, \ldots$ $\ldots, n$, ограниченными прямыми $x = 1\frac{1}{2}$, $2\frac{1}{2}, \ldots, n + \frac{1}{2}$ и осью x (см. рис. 7).

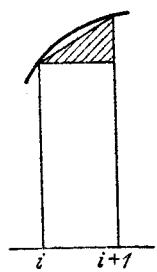


Рис. 6

Очевидно, что она составлена из треугольника с площадью $\frac{1}{8}$ и трапеций со средними линиями в точках $x = 2, 3, \ldots$ $\ldots, n$, площади которых равны соответственно $\ln 2, \ldots, \ln n$. Очевидно,

$$\frac{1}{8} + \sum_{i=1}^{n} \ln i > \int_1^n \ln x \, dx + \frac{1}{2} \ln n.$$

Отсюда

$$\ln n! > n \ln n - n + 1 + \frac{1}{2} \ln n - \frac{1}{8} = \left(n + \frac{1}{2}\right) \ln n - n + \frac{7}{8}.$$

Таким образом,

$$\left(n + \frac{1}{2}\right) \ln n - n + \frac{7}{8} < \ln n! < \left(n + \frac{1}{2}\right) \ln (n+1) - n.$$

Разделив на $\left(n+\frac{1}{2}\right)\ln n$, получим (см. также (3))

$$1-\frac{n-\frac{7}{8}}{\left(n+\frac{1}{2}\right)\ln n}<\frac{\ln n!}{\left(n+\frac{1}{2}\right)\ln n}<$$

$$<1+\frac{\left(n+\frac{1}{2}\right)\ln\left(1+\frac{1}{n}\right)-n}{\left(n+\frac{1}{2}\right)\ln n},$$

т. е.

$$\frac{\ln n!}{\left(n+\frac{1}{2}\right)\ln n}=1+O\left(\frac{1}{\ln n}\right).$$

Теорема доказана.

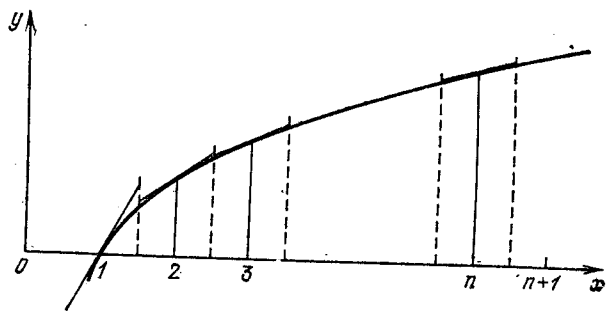


Рис. 7

Замечание. Из доказательства теоремы вытекает также, что

$$e^{7/8}\sqrt{n}\,n^n e^{-n}<n!<\sqrt{n+1}\,n^n e^{-n}\left(1+\frac{1}{n}\right)^n\sim e\sqrt{n}\,n^n e^{-n},$$

т. е.

$$n!\asymp\sqrt{n}\,n^n e^{-n}. \tag{31}$$

Асимптотика $n!$

Теорема 9 (формула Стирлинга). $n!\sim a\sqrt{n}\,n^n e^{-n}$.

Доказательство. Положим

$$a_n = \frac{n!}{\sqrt{n}\, n^n e^{-n}}.$$

Рассмотрим отношение

$$\frac{a_{n+1}}{a_n} = \frac{(n+1)!}{\sqrt{n+1}\,(n+1)^{n+1}e^{-(n+1)}} \frac{\sqrt{n}\, n^n e^{-n}}{n!} = \frac{e}{\left(1+\frac{1}{n}\right)^{n+1/2}}.$$

Учитывая (4), получаем

$$0 \leqslant \frac{a_{n+1}}{a_n} < 1.$$

Значит, последовательность $\{a_n\}$ убывающая и ограниченная снизу, и поэтому она сходится к некоторому числу a. В силу неравенств (31)

$$e^{7/8} \leqslant a \leqslant e.$$

В частности, $a \neq 0$. Следовательно, установлена асимптотика

$$n! \sim a\sqrt{n}\, n^n e^{-n}.$$

Для вычисления константы a нам понадобится формула Валлиса, которая является содержанием следующей теоремы.

Теорема 10. $\sqrt{\pi} = \lim\limits_{n\to\infty} \frac{1}{\sqrt{n}} \frac{2^{2n}(n!)^2}{(2n)!}$.

Доказательство. Возьмем $\int\limits_0^{\pi/2} \sin^n x\, dx\ (n \geqslant 2)$ и проинтегрируем его по частям:

$$\int_0^{\pi/2} \sin^n x\, dx =$$

$$= -\sin^{n-1} x \cos x \Big|_0^{\pi/2} + (n-1) \int_0^{\pi/2} \sin^{n-2} x \cos^2 x\, dx =$$

$$= (n-1) \int_0^{\pi/2} \sin^{n-2} x\, dx - (n-1) \int_0^{\pi/2} \sin^n x\, dx.$$

Разрешая полученное равенство относительно $\int\limits_0^{\pi/2} \sin^n x\, dx$,

получим

$$\int_0^{\pi/2} \sin^n x\,dx = \frac{n-1}{n} \int_0^{\pi/2} \sin^{n-2} x\,dx.$$

Данное рекуррентное соотношение дает при $n = 2k$

$$\int_0^{\pi/2} \sin^{2k} x\,dx = \frac{2k-1}{2k} \int_0^{\pi/2} \sin^{2k-2} x\,dx = \ldots,$$

$$\ldots = \frac{2k-1}{2k} \frac{2k-3}{2k-2} \cdots \frac{1}{2} \int_0^{\pi/2} dx = \frac{\pi}{2} \frac{(2k-1)(2k-3)\ldots 1}{2k(2k-2)\ldots 2}$$

и при $n = 2k+1$

$$\int_0^{\pi/2} \sin^{2k+1} x\,dx = \frac{2k}{2k+1} \int_0^{\pi/2} \sin^{2k-1} x\,dx = \ldots,$$

$$\ldots = \frac{2k}{2k+1} \frac{2k-2}{2k-1} \cdots \frac{2}{3} \int_0^{\pi/2} \sin x\,dx = \frac{2k(2k-2)\ldots 2}{(2k+1)(2k-1)\ldots 3}.$$

Отсюда получаем

$$\lambda_k = \int_0^{\pi/2} \sin^{2k} x\,dx \left(\int_0^{\pi/2} \sin^{2k+1} x\,dx \right)^{-1} =$$

$$= \frac{\pi}{2}(2k+1) \left[\frac{(2k-1)(2k-3)\ldots 1}{2k(2k-2)\ldots 2} \right]^2.$$

Поскольку в процессе интегрирования $0 \leqslant x \leqslant \pi/2$, то $0 \leqslant \sin x \leqslant 1$. Мы имеем

$$\sin^{2k+1} x \leqslant \sin^{2k} x \leqslant \sin^{2k-1} x$$

и

$$\int_0^{\pi/2} \sin^{2k+1} x\,dx \leqslant \int_0^{\pi/2} \sin^{2k} x\,dx \leqslant \int_0^{\pi/2} \sin^{2k-1} x\,dx.$$

Оценим величину λ_k:

$$1 \leqslant \lambda_k = \int_0^{\pi/2} \sin^{2k} x\,dx \left(\int_0^{\pi/2} \sin^{2k+1} x\,dx \right)^{-1} \leqslant$$

$$\leqslant \int_0^{\pi/2} \sin^{2k-1} x\,dx \left(\int_0^{\pi/2} \sin^{2k+1} x\,dx \right)^{-1} = 1 + \frac{1}{2k}.$$

Отсюда следует, что $\lambda_k \to 1$ (при $k \to \infty$), и, значит,

$$\sqrt{\pi} = \lim_{k\to\infty} \frac{1}{\sqrt{k}} \frac{2k\,(2k-2)\,\ldots\,2}{(2k-1)\,(2k-3)\,\ldots\,1} =$$

$$= \lim_{k\to\infty} \frac{1}{\sqrt{k}} \frac{[2k\,(2k-2)\,\ldots\,2]^2}{(2k)!} = \lim_{k\to\infty} \frac{2^{2k}\,(k!)^2}{\sqrt{k}\,(2k)!}.$$

Теорема доказана.

Теперь остается вычислить константу a. Возьмем выражение

$$\frac{a_n^2}{a_{2n}\sqrt{2}} = \left(\frac{n!}{\sqrt{n}\,n^n e^{-n}}\right)^2 \frac{\sqrt{2n}\,(2n)^{2n}e^{-2n}}{(2n)!\sqrt{2}} = \frac{1}{\sqrt{n}} \frac{2^{2n}\,(n!)^2}{(2n)!}.$$

Мы имеем

$$\lim_{n\to\infty} \frac{a_n^2}{a_{2n}\sqrt{2}} = \frac{a}{\sqrt{2}},$$

а с другой стороны, в силу формулы Валлиса

$$\lim_{n\to\infty} \frac{a_n^2}{a_{2n}\sqrt{2}} = \sqrt{\pi}.$$

Отсюда $a = \sqrt{2\pi}$.

Асимптотика $\binom{n}{k}$.

1. При $k \to \infty$ и $n-k \to \infty$ из формулы Стирлинга получаем

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \sim \frac{\sqrt{n}}{\sqrt{2\pi k\,(n-k)}} \frac{n^n}{k^k\,(n-k)^{n-k}},$$

В частности, при четном n и $k = n/2$ имеем

$$\binom{n}{n/2} \sim \frac{2^n}{\sqrt{\pi\, n/2}} \quad \text{(вариант формулы Валлиса)}.$$

2. Если взять $k = [n/2] + r$, где $|r| \leqslant \alpha(n)\sqrt{n}$, то при определенном ограничении на рост $\alpha(n)$ можно асимптотическому выражению для $\binom{n}{k}$ придать более компактный вид.

Теорема 11. *При* $|r| \leqslant \alpha(n)\sqrt{n}$, *где* $\alpha(n) = o(n^{1/6})$, *имеет место соотношение*

$$\binom{n}{[n/2]+r} \sim \frac{2^n}{\sqrt{\pi\, n/2}}\, e^{-2r^2/n}, \qquad n \to \infty.$$

Доказательство. Положив $\frac{n}{2}+r'=\left[\frac{n}{2}\right]+r$, выполним преобразования

$$\ln k^k(n-k)^{n-k}=\left(\frac{n}{2}+r'\right)\ln\left(\frac{n}{2}+r'\right)+\left(\frac{n}{2}-r'\right)\ln\left(\frac{n}{2}-r'\right)=$$

$$=\left(\frac{n}{2}+r'\right)\ln\frac{n}{2}+\left(\frac{n}{2}+r'\right)\ln\left(1+\frac{2r'}{n}\right)+$$

$$+\left(\frac{n}{2}-r'\right)\ln\frac{n}{2}+\left(\frac{n}{2}-r'\right)\ln\left(1-\frac{2r'}{n}\right).$$

Множители вида $\ln(1+x)$ заменим куском ряда так, чтобы погрешность была $o(1)$. Для этого достаточно взять разложение

$$\ln(1+x)=x-\frac{x^2}{2}+O(x^3),$$

так как

$$nO\left(\frac{r'^3}{n^3}\right)=O\left(\frac{r'^3}{n^2}\right)=o(1).$$

Тогда, учитывая также, что $|r-r'|<1$ и $r=o(n^{2/3})$, получим

$$\ln k^k(n-k)^{n-k}=$$

$$=n\ln\frac{n}{2}+\left(\frac{n}{2}+r'\right)\ln\left(1+\frac{2r'}{n}\right)+\left(\frac{n}{2}-r'\right)\ln\left(1-\frac{2r'}{n}\right)=$$

$$=n\ln\frac{n}{2}+\left(\frac{n}{2}+r'\right)\left(\frac{2r'}{n}-\frac{1}{2}\frac{4r'^2}{n^2}\right)+$$

$$+\left(\frac{n}{2}-r'\right)\left(-\frac{2r'}{n}-\frac{1}{2}\frac{4r'^2}{n^2}\right)+o(1)=$$

$$=n\ln\frac{n}{2}+\frac{2r'^2}{n}+o(1)=n\ln\frac{n}{2}+\frac{2r^2}{n}+o(1).$$

Отсюда получаем

$$\binom{n}{[n/2]+r}\sim\frac{2^n}{\sqrt{\pi n/2}}e^{-2r^2/n},\qquad n\to\infty.$$

Теорема доказана.

Часто вместо величины r берут $z = 2r/\sqrt{n}$. Мы приходим к соотношению

$$\binom{n}{\left[\frac{n}{2}\right] + \frac{\sqrt{n}}{2} z} \sim \frac{2^n}{\sqrt{\pi n/2}} e^{-z^2/2}.$$

Асимптотика для суммы $\sum \binom{n}{k}$.

Сначала установим один результат, характеризующий распределение вершин n-мерного единичного куба по слоям.

Теорема 12. *Пусть $\alpha(n)$ — произвольная (сколь угодно медленно) растущая положительная функция. Тогда*

$$\frac{1}{2^n} \sum_{\left[\frac{n}{2}\right] - \alpha(n)\sqrt{n} \leqslant k \leqslant \left[\frac{n}{2}\right] + \alpha(n)\sqrt{n}} \binom{n}{k} \sim 1.$$

Доказательство. Пусть $k_0 \geqslant n/2$. Тогда

$$\sum_{i=k_0}^{n} \binom{n}{i} = \binom{n}{k_0} + \binom{n}{k_0+1} + \ldots + \binom{n}{n} < \binom{n}{k_0}\left(1 + \frac{n-k_0}{k_0+1} + \right.$$

$$\left. + \left(\frac{n-k_0}{k_0+1}\right)^2 + \ldots\right) < \binom{n}{k_0} \frac{1}{1 - \frac{n-k_0}{k_0+1}} <$$

$$< \binom{n}{k_0} \frac{1}{1 - \frac{n-k_0}{k_0}} = \binom{n}{k_0} \frac{k_0}{2k_0 - n}.$$

С другой стороны (см. рис. 8), из сравнения площадей имеем

$$\binom{n}{k_0}(2k_0 - n) < \binom{n}{n-k_0} + \binom{n}{n-k_0+1} + \ldots + \binom{n}{k_0} < 2^n.$$

Отсюда

$$\sum_{i=k_0}^{n} \binom{n}{i} < \frac{k_0 2^n}{(2k_0 - n)^2}.$$

Положим $k_0 = \left[\frac{n}{2}\right] + \left[\alpha(n)\sqrt{n}\right]$. Тогда

$$\frac{1}{2^n}\sum_{i=k_0}^{n}\binom{n}{i} \leqslant \frac{1}{8\alpha^2(n)} \to 0.$$

В силу симметрии

$$\frac{1}{2^n}\sum_{i=0}^{n-k_0}\binom{n}{i} \leqslant \frac{1}{8\alpha^2(n)} \to 0.$$

Отсюда, учитывая, что $\sum_{i=0}^{n}\binom{n}{i} = 2^n$, получаем требуемое.

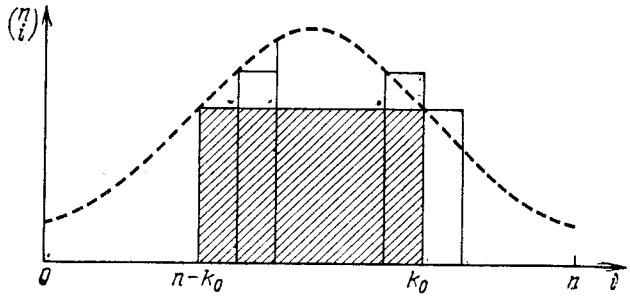


Рис. 8

Доказанная теорема утверждает, что почти все точки n-мерного единичного куба сосредоточены в слоях с номерами k такими, что

$$\left[\frac{n}{2}\right] - \alpha(n)\sqrt{n} \leqslant k \leqslant \left[\frac{n}{2}\right] + \alpha(n)\sqrt{n},$$

где $\alpha(n)$ — данная фиксированная растущая (сколь угодно медленно) функция, т. е. в слоях, «близких» к среднему слою.

Более точные оценки получаются, если использовать асимптотическое выражение для $\binom{n}{k}$.

Теорема 13. *Пусть* $a < b$. *Тогда*

$$\frac{1}{2^n} \sum_{\left[\frac{n}{2}\right]+\frac{a\sqrt{n}}{2} \leqslant k \leqslant \left[\frac{n}{2}\right]+\frac{b\sqrt{n}}{2}} \binom{n}{k} \sim \int_a^b e^{-z^2/2} dz.$$

Доказательство. Пользуясь теоремой 11 и полагая $z = \frac{2r}{\sqrt{n}} = \frac{2(k - [n/2])}{\sqrt{n}}$, получим

$$\frac{1}{2^n} \sum_{\left[\frac{n}{2}\right]+\frac{a\sqrt{n}}{2} \leqslant k \leqslant \left[\frac{n}{2}\right]+\frac{b\sqrt{n}}{2}} \binom{n}{k} \sim \frac{1}{2^n} \sum_{\left[\frac{n}{2}\right]+\frac{a\sqrt{n}}{2} \leqslant k \leqslant \left[\frac{n}{2}\right]+\frac{b\sqrt{n}}{2}} \frac{2^n e^{-z^2/2}}{\sqrt{\pi n/2}} =$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{\left[\frac{n}{2}\right]+\frac{a\sqrt{n}}{2} \leqslant k \leqslant \left[\frac{n}{2}\right]+\frac{b\sqrt{n}}{2}} e^{-z^2/2} \frac{2}{\sqrt{n}} =$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{\left[\frac{n}{2}\right]+\frac{a\sqrt{n}}{2} \leqslant k \leqslant \left[\frac{n}{2}\right]+\frac{b\sqrt{n}}{2}} e^{-z^2/2} \Delta z \to \frac{1}{\sqrt{2\pi}} \int_a^b e^{-z^2/2} dz,$$

так как $\Delta z = \frac{2(r+1)}{\sqrt{n}} - \frac{2r}{\sqrt{n}} = \frac{2}{\sqrt{n}}$. Теорема доказана.

Замечание. Доказательство теоремы может быть обобщено на случай, когда $a = -\alpha(n)$ и $b = \alpha(n)$ $(\alpha(n) \to \infty)$. Тогда

$$\frac{1}{2^n} \sum_{\left[\frac{n}{2}\right]-\alpha(n)\frac{\sqrt{n}}{2} \leqslant k \leqslant \left[\frac{n}{2}\right]+\alpha(n)\frac{\sqrt{n}}{2}} \binom{n}{k} \sim$$

$$\sim \frac{1}{\sqrt{2\pi}} \int_{-\alpha(n)}^{\alpha(n)} e^{-z^2/2} dz \sim \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-z^2/2} dz = 1.$$

Асимптотика $\Phi(n)$.

При нахождении асимптотики для $\Phi(n)$ можно исходить из представления $\Phi(n)$ в виде бесконечного ряда (формулы Добинского) или из выражения $\Phi(n)$ в виде многочлена (30). Запишем формулу (30) несколько иначе:

$$\Phi(n) = \sum_{k=1}^{n} \left(1 - \frac{1}{1!} + \frac{1}{2!} - \ldots + (-1)^{n-k} \frac{1}{(n-k)!}\right) \frac{k^n}{k!}.$$

Возьмем две (не обязательно целочисленные) функции $a(n)$ и $b(n)$ таких, что

$$1 < a(n) < b(n) < n,$$
$$a(n) \to \infty, \quad n - b(n) \to \infty, \quad n \to \infty,$$

и разобьем сумму на три части:

$$\Phi(n) = \Phi_1(n) + \Phi_2(n) + \Phi_3(n),$$

где

$$\Phi_1(n) = \sum_{1 < k < a(n)} \left(1 - \frac{1}{1!} + \ldots + (-1)^{n-k} \frac{1}{(n-k)!}\right) \frac{k^n}{k!},$$

$$\Phi_2(n) = \sum_{a(n) \leqslant k \leqslant b(n)} \left(1 - \frac{1}{1!} + \ldots + (-1)^{n-k} \frac{1}{(n-k)!}\right) \frac{k^n}{k!},$$

$$\Phi_3(n) = \sum_{b(n) < k < n} \left(1 - \frac{1}{1!} + \ldots + (-1)^{n-k} \frac{1}{(n-k)!}\right) \frac{k^n}{k!}.$$

Поскольку $n - b(n) \to \infty$, то для $k \leqslant b(n)$ имеем $n - k \geqslant n - b(n)$ и

$$1 - \frac{1}{1!} + \ldots + (-1)^{n-k} \frac{1}{(n-k)!} \sim \frac{1}{e}.$$

Мы получаем

$$\Phi_1(n) \sim \frac{1}{e} \sum_{1 < k < a(n)} \frac{k^n}{k!},$$

$$\Phi_2(n) \sim \frac{1}{e} \sum_{a(n) \leqslant k \leqslant b(n)} \frac{k^n}{k!},$$

$$\Phi_3(n) < \sum_{b(n) < k < n} \left(1 + \frac{1}{1!} + \ldots + \frac{1}{(n-k)!}\right) \frac{k^n}{k!} < e \sum_{b(n) < k < n} \frac{k^n}{k!}.$$

Во всех суммах присутствует член вида $k^n/k!$ Покажем, что последовательность $\{k^n/k!\}$ унимодальна. Сравним два соседних члена $k^n/k!$ и $(k+1)^n/(k+1)!$ Легко видеть, что

$$\frac{k^n}{k!} \gtreqless \frac{(k+1)^n}{(k+1)!} \Leftrightarrow n \gtreqless \frac{\ln(k+1)}{\ln\left(1 + \frac{1}{k}\right)}$$

(данная запись — сокращенная запись для трех условий).

Поскольку функция $\dfrac{\ln(k+1)}{\ln\left(1+\frac{1}{k}\right)}$ монотонно возрастающая по k, то в случае:

а) $\dfrac{k^n}{k!} < \dfrac{(k+1)^n}{(k+1)!}$ имеем $n > \dfrac{\ln(k+1)}{\ln\left(1+\frac{1}{k}\right)} > \dfrac{\ln k}{\ln\left(1+\frac{1}{k-1}\right)}$,

т. е. $n > \dfrac{\ln k}{\ln\left(1+\frac{1}{k-1}\right)}$ $\left(\text{поэтому } \dfrac{(k-1)^n}{(k-1)!} < \dfrac{k^n}{k!} \text{ и т. д.}\right)$;

б) $\dfrac{k^n}{k!} > \dfrac{(k+1)^n}{(k+1)!}$ имеем $n < \dfrac{\ln(k+1)}{\ln\left(1+\frac{1}{k}\right)} < \dfrac{\ln(k+2)}{\ln\left(1+\frac{1}{k+1}\right)}$,

т. е.

$n < \dfrac{\ln(k+2)}{\ln\left(1+\frac{1}{k+1}\right)}$ $\left(\text{поэтому } \dfrac{(k+1)^n}{(k+1)!} > \dfrac{(k+2)^n}{(k+2)!} \text{ и т. д.}\right)$;

в) $\dfrac{k^n}{k!} = \dfrac{(k+1)^n}{(k+1)!}$ из строгой монотонности $\dfrac{\ln(k+1)}{\ln\left(1+\frac{1}{k}\right)}$

следует, что $\dfrac{(k-1)^n}{(k-1)!} < \dfrac{k^n}{k!}$ и $\dfrac{(k+1)^n}{(k+1)!} > \dfrac{(k+2)^n}{(k+2)!}$ и т. д.

Обозначим через k_0 наибольшее значение k, для которого $k^n/k!$ максимально. Произведем оценку числа k_0. Очевидно,

$$\frac{(k_0-1)^n}{(k_0-1)!} \leqslant \frac{k_0^n}{k_0!} > \frac{(k_0+1)^n}{(k_0+1)!},$$

$$n \ln\left(1+\frac{1}{k_0-1}\right) \geqslant \ln k_0,$$

$$n \ln\left(1+\frac{1}{k_0}\right) < \ln(k_0+1).$$

Так как (см. (3))

$$\ln\left(1+\frac{1}{k_0-1}\right) < \frac{1}{k_0-1},$$

то

$$n > (k_0-1)\ln k_0 > (k_0-1)\ln(k_0-1),$$

а так как (см. (4))

$$\ln\left(1+\frac{1}{k_0}\right)>\frac{1}{k_0+\frac{1}{2}},$$

то

$$n<\left(k_0+\frac{1}{2}\right)\ln(k_0+1)<(k_0+1)\ln(k_0+1).$$

Поэтому из монотонности функции $x\ln x$ следует, что k_0 отличается от единственного решения r уравнения

$$n=x\ln x$$

менее чем на 1, т. е.

$$|r-k_0|<1$$

или в силу целочисленности k_0

$$k_0=[r] \quad \text{либо} \quad k_0=]r[.$$

Положим далее $a(n)=r-\sqrt{n}$ и $b(n)=r+\sqrt{n}$. В силу того что $r\sim\frac{n}{\ln n}(1+o(1))$, при достаточно больших n будет

$$1<a(n)<b(n)<n.$$

Условия $a(n)\to\infty$ и $n-b(n)\to\infty$ также выполнены.

Сначала займемся изучением

$$\Phi_2(n)\sim\frac{1}{e}\sum_{a(n)<k<b(n)}\frac{k^n}{k!}.$$

Заменим в $\Phi_2(n)$ члены $k^n/k!$ их асимптотическими выражениями, получающимися по формуле Стирлинга:

$$\frac{k^n}{k!}\sim\frac{1}{\sqrt{2\pi}}k^{n-k-\frac{1}{2}}e^k(1+o(1)),$$

и положим $s=k-r$ (здесь параметр s может стать не целочисленным, но $\Delta s=\Delta k=1$). Тогда получим

$$\sum_{a(n)<k<b(n)}\frac{k^n}{k!}\sim\frac{1}{\sqrt{2\pi}}\sum_{r-\sqrt{n}<k<r+\sqrt{n}}k^{n-k-\frac{1}{2}}e^k(1+o(1))=$$

$$=\frac{1}{\sqrt{2\pi}}\sum_{-\sqrt{n}<s<\sqrt{n}}(r+s)^{n-r-s-\frac{1}{2}}e^{r+s}(1+o(1)).$$

Заметим, далее, что

$$r+s=\exp\{\ln(r+s)\}=\exp\left\{\ln r\left(1+\frac{s}{r}\right)\right\}=$$
$$=\exp\left\{\ln r+\ln\left(1+\frac{s}{r}\right)\right\}.$$

Учитывая соотношение $\left|\frac{s}{r}\right|=O\left(\frac{\ln n}{\sqrt{n}}\right)$, разложим в ряд $\ln\left(1+\frac{s}{r}\right)$ и оставим в разложении столько членов, чтобы остаток, умноженный на $\left(n-r-s-\frac{1}{2}\right)\sim n$, был бы равен $o(1)$. Для этого достаточно взять два первых члена разложения, т. е.

$$\ln\left(1+\frac{s}{r}\right)=\frac{s}{r}-\frac{1}{2}\frac{s^2}{r^2}+O\left(\frac{s^3}{r^3}\right),$$

ибо

$$nO\left(\frac{s^3}{r^3}\right)=O\left(\frac{\ln^3 n}{\sqrt{n}}\right)=o(1).$$

Таким образом, мы получаем

$$(r+s)^{n-r-s-\frac{1}{2}}e^{r+s}=$$
$$=\exp\left\{\left(\ln r+\frac{s}{r}-\frac{1}{2}\frac{s^2}{r^2}\right)\left(n-r-s-\frac{1}{2}\right)+r+s+o(1)\right\}.$$

Произведем далее преобразование показателя:

$$\left(\ln r+\frac{s}{r}-\frac{1}{2}\frac{s^2}{r^2}\right)\left(n-r-s-\frac{1}{2}\right)+r+s=$$
$$=n\ln r+\frac{ns}{r}-\frac{ns^2}{2r^2}-r\ln r-s+\frac{s^2}{2r}-s\ln r-$$
$$-\frac{s^2}{r}+\frac{s^3}{2r^2}-\frac{1}{2}\ln r-\frac{s}{2r}+\frac{s^2}{4r^2}+r+s.$$

В нем, в силу соотношения $r\ln r=n$, выполняется равенство $\frac{ns}{r}=s\ln r$, а члены $\frac{s^3}{2r^2}$, $\frac{s}{2r}$ и $\frac{s^2}{4r^2}$ равны $o(1)$.

Продолжив преобразование, получим

$$n\left(\ln r+\frac{1}{\ln r}-1\right)-\frac{1}{2}\ln r-\frac{s^2}{2r}\left(\frac{n}{r}+1\right)+o(1).$$

В результате получаем

$$\Phi_2(n)=\frac{\exp\left\{n\left(\ln r+\frac{1}{\ln r}-1\right)-1\right\}}{\sqrt{\frac{n}{r}+1}}\frac{1}{\sqrt{2\pi}}\times$$

$$\times\sum_{-\sqrt{n}\leqslant s\leqslant\sqrt{n}}\exp\left\{-\frac{s^2}{2r}\left(\frac{n}{r}+1\right)\right\}\sqrt{\frac{\frac{n}{r}+1}{r}}(1+o(1)).$$

В последней сумме произведем замену переменной, взяв

$$z=s\sqrt{\frac{\frac{n}{r}+1}{r}},\quad \Delta z=\sqrt{\frac{\frac{n}{r}+1}{r}}=o(1),$$

и перейдем к интегралу:

$$\frac{\exp\left\{n\left(\ln r+\frac{1}{\ln r}-1\right)-1\right\}}{\sqrt{\frac{n}{r}+1}}\frac{1}{\sqrt{2\pi}}\int\limits_{-\sqrt{\frac{n}{r}\left(\frac{n}{r}+1\right)}}^{+\sqrt{\frac{n}{r}\left(\frac{n}{r}+1\right)}}e^{-z^2/2}dz(1+o(1)).$$

В силу соотношения $\sqrt{\frac{n}{r}\left(\frac{n}{r}+1\right)}\sim\frac{n}{r}\sim\ln n$ интеграл, деленный на $\sqrt{2\pi}$, стремится к 1, и поэтому

$$\Phi_2(n)\sim\frac{\exp\left\{n\left(\ln r+\frac{1}{\ln r}-1\right)-1\right\}}{\sqrt{\frac{n}{r}+1}}\sim\frac{r^n e^{r-n-1}}{\sqrt{\ln n}}.$$

Суммы $\Phi_1(n)$ и $\Phi_3(n)$ оцениваются стандартным образом: каждый член заменяется на максимальный, который находится из условия того, что $\{k^n/k!\}$ унимодальна и ее максимум достигается в интервале $(r-\sqrt{n},\ r+\sqrt{n})$.

Поэтому

$$\Phi_1(n) \sim \frac{1}{e} \sum_{k=1}^{<r-\sqrt{n}} \frac{k^n}{k!} = \frac{1}{e} \sum_{k=1}^{k_1} \frac{k^n}{k!} < \frac{1}{e} \frac{k_1^n}{k_1!} (r - \sqrt{n}) \leqslant$$

$$\leqslant \frac{r}{e} \frac{\exp\left\{n\left(\ln r + \frac{1}{\ln r} - 1\right)\right\}}{\sqrt{r}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{n}{r}\left(\frac{n}{r}+1\right)\right\} =$$

$$= \frac{\exp\left\{n\left(\ln r + \frac{1}{\ln r} - 1\right) - 1\right\}}{\sqrt{\frac{n}{r}+1}} \frac{\sqrt{r}\sqrt{\frac{n}{r}+1}}{\sqrt{2\pi}} \times$$

$$\times \exp\left\{-\frac{1}{2}\frac{n}{r}\left(\frac{n}{r}+1\right)\right\} \sim \Phi_2(n)\sqrt{\frac{n}{2\pi}} \exp\left\{-\frac{1}{2}\frac{n}{r}\left(\frac{n}{r}+1\right)\right\} =$$

$$= o(\Phi_2(n)).$$

Аналогично

$$\Phi_3(n) < e \sum_{k>r+\sqrt{n}}^{n} \frac{k^n}{k!} = e \sum_{k=k_2}^{n} \frac{k^n}{k!} < e \frac{k_2^n}{k_2!} (n - r - \sqrt{n}) \leqslant$$

$$\leqslant ne \frac{\exp\left\{n\left(\ln r + \frac{1}{\ln r} - 1\right)\right\}}{\sqrt{r}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{n}{r}\left(\frac{n}{r}+1\right)\right\} =$$

$$= \frac{\exp\left\{n\left(\ln r + \frac{1}{\ln r} - 1\right) - 1\right\}}{\sqrt{\frac{n}{r}+1}} \frac{e^2}{\sqrt{2\pi}} \frac{n\sqrt{\frac{n}{r}+1}}{\sqrt{r}} \times$$

$$\times \exp\left\{-\frac{1}{2}\frac{n}{r}\left(\frac{n}{r}+1\right)\right\} \sim \Phi_2(n) \frac{e^2}{\sqrt{2\pi}} \sqrt{n} \ln n \times$$

$$\times \exp\left\{-\frac{1}{2}\frac{n}{r}\left(\frac{n}{r}+1\right)\right\} = o(\Phi_2(n)).$$

Таким образом, мы доказали теорему.

Теорема 14. $\Phi(n) \sim \frac{r^n e^{r-n-1}}{\sqrt{\ln n}}$, *где* r — *корень уравнения* $x \ln x = n$.

ЧАСТЬ III
ГРАФЫ И СЕТИ

Теория графов и сетей может быть отнесена к конечной геометрии. Геометрическая интуиция играет в ней существенную роль как в предвидении, так и в получении результатов.

В данной части содержатся некоторые факты из трех направлений: проблемы реализуемости одного класса объектов в другом, метрические вопросы, касающиеся графов и сетей, и структурные особенности этих объектов.

Глава 1
ГРАФЫ

§ 1. Реализация в евклидовом пространстве. Изоморфизм

В дальнейшем мы часто будем пользоваться понятиями «множество» и «набор». Термин «множество» имеет общепринятый смысл. Под «набором» здесь мы понимаем неупорядоченную систему объектов из некоторого множества, в которой один и тот же объект может встречаться несколько раз.

Определение. Множество $\mathfrak{M} = \{a_1, a_2, \ldots\}$ и набор $\mathfrak{N}$ неупорядоченных пар объектов (a_{i_k}, a_{j_k}) из $\mathfrak{M}$ называется *графом* Г. Объекты множества $\mathfrak{M}$ называются *вершинами графа*, а объекты набора $\mathfrak{N}$ — *ребрами графа*. Про ребра (a_i, a_j) будем говорить, что они соединяют вершины a_i и a_j.

Пример 1. Пусть $\mathfrak{M} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, $\mathfrak{N} = \{(a_1, a_2), (a_2, a_2), (a_4, a_5), (a_5, a_6), (a_5, a_6), (a_6, a_7), (a_5, a_7)\}$. Тогда $\mathfrak{M}$ и $\mathfrak{N}$ определяют граф.

В случае, если множество $\mathfrak{M}$ и набор $\mathfrak{N}$ состоят из конечного числа объектов и пар, то граф Г называется *конечным*.

Пусть a_i и a_j — произвольные вершины графа Г.

Определение. Система ребер графа Г

$$A_{a_ia_j} = \{(a_{i_1}, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{s-1}}, a_{i_s})\},$$

где $a_{i_1} = a_i$ и $a_{i_s} = a_j$, называется *путем, соединяющим вершины* a_i *и* a_j. Для любого ребра, принадлежащего пути $A_{a_ia_j}$, мы будем говорить, что *путь* $A_{a_ia_j}$ *проходит через* это *ребро*. Аналогично, если вершина a принадлежит некоторому ребру пути $A_{a_ia_j}$, то говорим, что *путь* $A_{a_ia_j}$ *проходит через вершину* a.

Определение. Путь $A_{a_ia_j}$, не проходящий дважды через одно ребро, называется *циклом*, если $a_i = a_j$. В частности, цикл $\{(a_i, a_i)\}$ будем называть *петлей*.

Определение. Граф Г называется *связным*, если для любых двух различных вершин a_i и a_j графа Г существует путь, соединяющий эти вершины.

Легко видеть, что граф из примера 1 является конечным, несвязным и содержащим петли. (Ясно, что связный граф не содержит «изолированных» вершин, т. е. каждая его вершина принадлежит по крайней мере одному его ребру.)

Введенное нами понятие графа является весьма абстрактным. Оно родственно понятию одномерного комплекса в топологии. Для последующих рассмотрений желательно иметь какую-либо наглядную интерпретацию графа. С этой целью будем рассматривать в евклидовом пространстве фигуры определенного вида. Каждая из таких фигур Γ состоит из различных вершин $b_1, b_2, \ldots$ и кривых (являющихся либо дугами окружностей, либо отрезками прямых), каждая из которых соединяет некоторые пары вершин (b_i, b_j) (возможно и вырождение $b_i = b_j$). Мы предполагаем, что никакая внутренняя точка кривой, принадлежащей фигуре Γ, не является вершиной или внутренней точкой другой кривой.

Определение. Фигура Γ называется *геометрической реализацией графа* Г, если существует взаимно однозначное соответствие между вершинами фигуры Γ и вершинами графа Г, а также между кривыми фигуры Γ и ребрами графа Г такое, что если $(b_{n_i}, b_{n_j}) \leftrightarrow (a_i, a_j)$, то $b_{n_i} \leftrightarrow a_i$, $b_{n_j} \leftrightarrow a_j$. (Соответствующие кривые и ребра соединяют соответствующие вершины.)

Пример 2. Фигура ε на рис. 1, как нетрудно убедиться, является геометрической реализацией графа Г из примера 1.

Спрашивается, любой ли граф Г можно реализовать в евклидовом пространстве? Если этот вопрос решается

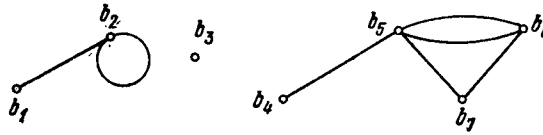


Рис. 1

положительно, то интересно знать, существует ли такое число ρ, что всякий граф допускает реализацию в евклидовом пространстве размерности ρ? В последнем случае желательно знать минимальное значение ρ. Ответ на все эти вопросы для случая конечных графов дают теорема 1 и пример 3. Теорема 1 представляет некоторую перефразировку известной теоремы из топологии.

Теорема 1. *Каждый конечный граф* Г *можно реализовать в трехмерном евклидовом пространстве.*

Доказательство. Пусть граф Г содержит m вершин и h ребер. Возьмем прямую и через нее проведем связку из h плоскостей. На прямой выберем m точек $b_1, b_2, \ldots, b_m$, сопоставим их вершинам графа соответственно $a_1, a_2, \ldots, a_m$. Каждому ребру графа Г поставим в соответствие плоскость из пучка. Пусть (a_i, a_j) — ребро графа Г. В плоскости, соответствующей ребру (a_i, a_j), соединим вершины b_i и b_j дугой окружности. Выполним такое построение для всех ребер графа Г, получим фигуру ε. Очевидно, что ε является геометрической реализацией графа Г.

Можно показать, что данная теорема не допускает усиления в направлении понижения размерности евклидова пространства, в котором может быть реализован любой граф.

Пример 3. На рис. 2 изображены два графа. Первый из них связан с решением известной задачи о трех домах и трех колодцах*). Второй — полный**) граф с пятью

*) Задача о трех домах и трех колодцах состоит в следующем: требуется от каждого дома проложить тропинку к каждому колодцу и так, чтобы тропинки не пересекались друг с другом.

**) То есть граф, содержащий все ребра вида (a_i, a_j), $1 \leqslant i < j \leqslant m$.

вершинами. В топологии доказывается, что данные графы не допускают реализации на плоскости. Доказательства этих фактов мы не приводим.

Интересный результат, выясняющий условия плоской реализуемости, был получен Л. С. Понтрягиным [28] и

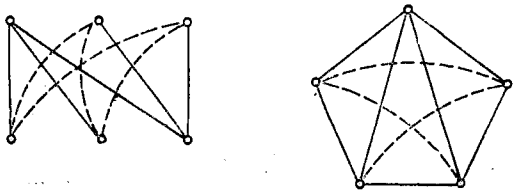


Рис. 2

независимо, но позже Куратовским. Ниже (теорема 2) приводится без доказательства формулировка этого факта.

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

С точки зрения этого понятия абстрактный граф и его геометрическая реализация являются изоморфными графами. В силу теоремы 1 вместо абстрактных конечных графов можно рассматривать их реализации. Другими словами, с графами можно обращаться как с геометрическими объектами.

Введем операцию *подразделения ребра графа* Γ. Пусть (a_i, a_j) — произвольное ребро графа Γ и a — объект, не принадлежащий $\mathfrak{M}$. Операция подразделения ребра (a_i, a_j) графа Γ состоит в построении графа Γ', имеющего своими вершинами множество

$$\mathfrak{M}' = \mathfrak{M} \cup \{a\},$$

содержащего все ребра графа Γ, кроме выделенного (a_i, a_j), и плюс два новых ребра (a_i, a), (a, a_j), т. е.

$$\mathfrak{R}' = (\mathfrak{R} \setminus (a_i, a_j)) \cup \{(a_i, a), (a, a_j)\}.$$

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

Определение. Графы Γ_1 и Γ_2 называются *гомеоморфными*, если существуют такие их подразделения, которые изоморфны.

Пример 4. На рис. 3 изображены два графа Γ_1 и Γ_2. Эти графы не изоморфны и в то же время гомеоморфны, так как каждый из них может быть подразделен до графа Γ, изображенного на рис. 4.

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

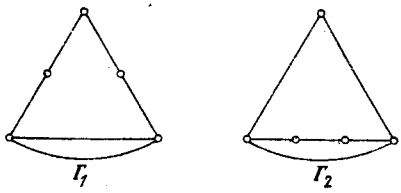


Рис. 3

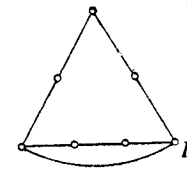


Рис. 4

Теорема 2 (критерий плоской реализуемости). *Для того чтобы конечный граф* Γ *имел плоскую реализацию, необходимо и достаточно, чтобы любой его подграф не был гомеоморфен ни одному из графов рис.* 2.

§ 2. Оценка числа графов

Пусть $\gamma(h)$ — максимальное число попарно неизоморфных графов без изолированных вершин с h ребрами.

Теорема 3. $\gamma(h) < c_1 (c_2 h)^h$, *где* c_1 *и* c_2 — *константы.*

Доказательство. Очевидно, что граф с h ребрами имеет не более $2h$ вершин. Занумеруем вершины графа натуральными числами 1, 2, ... Очевидно, что число сортов ребер, т. е. пар вершин, которые могут связываться ребрами, не превосходит величины

$$r = H_{2h}^2 = \binom{2h+1}{2} = h(2h+1).$$

Поскольку $\gamma(h)$ не превосходит максимального числа занумерованных попарно неизоморфных графов с h ребрами, а это число не больше, чем число сочетаний с повторениями из r элементов по h, то

$$\gamma(h) \leqslant H_r^h = \binom{r+h-1}{h} \leqslant$$

$$\leqslant \frac{(r+h-1)^h}{h!} < \frac{(2h^2+2h)^h}{\left(\frac{h}{e}\right)^h} = \left(1+\frac{1}{h}\right)^h (2eh)^h < e\,(2eh)^h,$$

что и требовалось доказать.

Следствие. *Максимальное число занумерованных попарно неизоморфных графов без изолированных вершин с h ребрами не превосходит $c_1(c_2h)^h$.*

Мы не делаем здесь оценок снизу для $\gamma(h)$ и потому лишены возможности понять качество полученной верхней оценки величины $\gamma(h)$.

Глава 2

СЕТИ

§ 1. Сети и их свойства

Обобщим понятие графа.

Определение. Множество $\mathfrak{M} = \{a_1, a_2, \ldots\}$ и набор $\mathfrak{N} = \{E_0; E_1, E_2, \ldots\}$, в котором каждое E_i есть набор элементов из $\mathfrak{M}$, т. е. $E_i = (a_{v_1(i)}, a_{v_2(i)}, \ldots)$, называется *сетью* и обозначается $\mathfrak{M}(E_0; E_1, E_2, \ldots)$. Объекты множества $\mathfrak{M}$ называются *вершинами*, а объекты из набора E_0 — *полюсами сети* *).

Пример 1. Пусть

$$\mathfrak{M} = \{1, 2, 3, 4, 5, 6, 7\}, \quad \mathfrak{N} = \{E_0; E_1, E_2, E_3, E_4, E_5\},$$

где

$$E_0 = (1, 2, 6), \quad E_1 = (1, 3, 3, 4, 5),$$

$$E_2 = (4, 4, 4, 5, 6), \quad E_3 = E_4 = (2, 4), \quad E_5 = (2, 5, 6, 7).$$

Тогда $\mathfrak{M}(E_0; E_1, E_2, E_3, E_4, E_5)$ будет сетью.

В случае, когда множество $\mathfrak{M}$ и набор $\mathfrak{N}$ конечны, сеть будет называться *конечной*. Так, например, только что рассмотренная сеть будет конечной. Сеть, в которой бесконечно по крайней мере $\mathfrak{M}$ или $\mathfrak{N}$, называется *бесконечной*. Частным случаем бесконечных сетей являются счетные сети, т. е. бесконечные сети, у которых $\mathfrak{M}$ и $\mathfrak{N}$ не более чем счетны.

Подобно тому как это было сделано в случае графов, можно ввести понятие геометрической реализации для конечной или счетной сети. Прежде всего введем одно

*) В литературе встречаются близкие понятия: понятие «блок-схемы» и понятие «гиперграфа». Понятие блок-схемы уже, чем понятие сети; а понятие гиперграфа отличается от понятий блок-схемы и сети незначительной деталью. Понятие блок-схемы появилось раньше понятия сети, а понятие гиперграфа — позже. См., например, [7, 32].

обозначение. Если E — набор, то через $\langle E\rangle$ будем обозначать множество всех объектов из E. Пусть $\mathfrak{M}(E_0; E_1, \ldots)$ — сеть. Разобьем множество $\mathfrak{M}$ на три непересекающиеся части:

$\mathfrak{M}_1 = \langle E_0\rangle$ — множество полюсов,

$\mathfrak{M}_2 = \mathfrak{M} \setminus \bigcup_{i\geqslant 0} \langle E_i\rangle$ — множество изолированных вершин, отличных от полюсов,

$\mathfrak{M}_3$ — прочие вершины.

Каждой вершине из множеств $\mathfrak{M}_1$ и $\mathfrak{M}_2$ сопоставим в трехмерном евклидовом пространстве точку так, чтобы различным вершинам соответствовали различные точки. Эти точки мы пометим символами соответствующих вершин a_i из $\mathfrak{M}$. Очевидно, полюсам будут соответствовать точки, помеченные символами $a_{\nu_1(0)}, a_{\nu_2(0)}, \ldots$ Каждому набору $E_i = (a_{\nu_1(i)}, a_{\nu_2(i)}, \ldots)$ из $\mathfrak{N}$ $(i \geqslant 1)$ сопоставим в трехмерном евклидовом пространстве круг (если E_i содержит

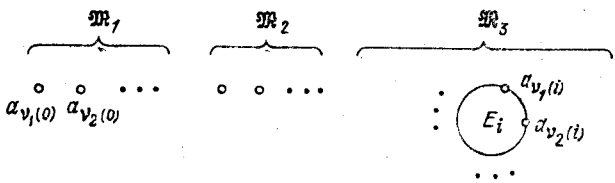


Рис. 1

один или два объекта, то можно вместо круга брать вершину или дугу), на периферии которого выбраны попарно различные вершины, помеченные символами $a_{\nu_1(i)}, a_{\nu_2(i)}, \ldots$ из набора E_i. При этом мы требуем, чтобы круги попарно не пересекались и не содержали выбранных ранее вершин (рис. 1).

Затем все точки, которые помечены одним и тем же символом a_i из $\mathfrak{M}$, соединяются связной компонентой A_i. Дополнительно потребуем, чтобы связная компонента A_i с построенными ранее вершинами и кругами имела общими только точки, помеченные символом a_i, и чтобы связные компоненты A_i и A_j $(i \neq j)$ не имели общих точек. Данную фигуру будем называть *геометрической реализацией исходной сети*. Очевидно, что образами вершин a_i из $\mathfrak{M}_2$ сети $\mathfrak{M}(E_0; E_1, \ldots)$ будут изолированные вершины

a_i; образами вершин a_i из $\mathfrak{M}_1$ и $\mathfrak{M}_3$ будут либо изолированные вершины a_i (если символ a_i встречается один раз и только в одном наборе), либо связная компонента A_i — в остальных случаях; образами наборов E_i $(i \geqslant 1)$ будут круги (соответственно вершины, дуги).

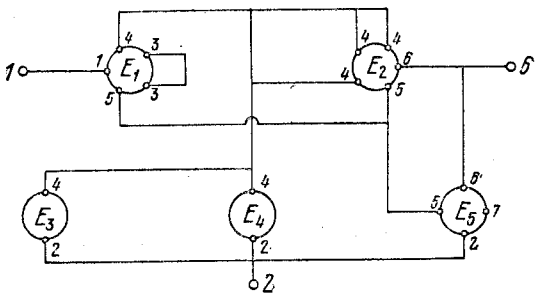


Рис. 2

Можно показать, что для каждой счетной сети существует геометрическая реализация.

Пример 2. Построив геометрическую реализацию сети из примера 1, мы получим фигуру, изображенную на рис. 2.

Данная фигура напоминает схему радиоприемника, из которой удалены все элементы: лампы, емкости, индуктивности и т. п.

Определение. Сети $\mathfrak{M}'(E'_0; E'_1, E'_2, \ldots)$ и $\mathfrak{M}''(E''_0; E''_1, E''_2, \ldots)$ называются *изоморфными*, если можно установить взаимно однозначное соответствие между объектами множеств $\mathfrak{M}'$ и $\mathfrak{M}''$, а также между объектами из наборов $\mathfrak{N}'$ и $\mathfrak{N}''$ так, что:

1) соответствующие наборы E' и E'' состоят из соответствующих объектов (с учетом кратности их вхождений);

2) наборы E'_0 и E''_0 соответствуют друг другу.

Очевидно, что абстрактная сеть изоморфна своей геометрической реализации. Поскольку нас будут интересовать сети с точностью до изоморфизма, то вместо абстрактных сетей можно рассматривать их геометрические реализации. В этом смысле сети представляют собой геометрические объекты.

Рассмотрим теперь некоторые классы сетей.

Очевидно, что класс сетей, у которых $E_0 = \Lambda$ и каждый набор E_i $(i \geqslant 1)$ состоит из двух объектов множества $\mathfrak{M}$, совпадает с классом графов.

Другой класс сетей дают так называемые деревья. *Деревом* называется конечный связный граф с выделен-

Рис. 3

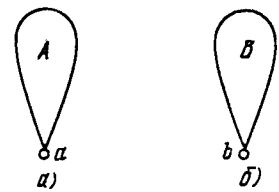


Рис. 4

ной вершиной, именуемой корнем, не содержащий циклов.

Очевидно, что дерево — однополюсная сеть, т. е. $E_0 = (a)$.

Приведем другое определение дерева, эквивалентное первому и основанное на индукции. Проще всего воспользоваться определением в геометрической форме.

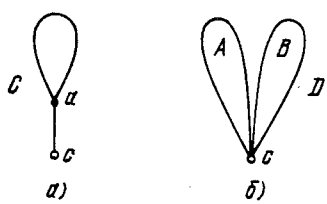


Рис. 5

Рис. 6

Базис индукции. Фигура, изображенная на рис. 3, является деревом с корнем a.

Индуктивный переход. Пусть A (рис. 4, $а$) — дерево с корнем a и B (рис. 4, $б$) — дерево с корнем b. Тогда фигура C (рис. 5, $а$), полученная из A «подключением» к корню a нового ребра, будет деревом с корнем c. Далее, фигура D (рис. 5, $б$), полученная из A и B путем объединения корней, будет деревом с корнем c, где $c = a = b$.

Легко видеть, что это индуктивное определение дерева можно сформулировать и в терминах абстрактной сети.

Базис индукции. Сеть $\mathfrak{M}(E_0; E_1)$, где $\mathfrak{M} = \{a, a_1\}$, $E_0 = (a)$, $E_1 = (a, a_1)$, является деревом с корнем a.

Индуктивный переход. Пусть $A = \mathfrak{M}_1(E_0'; E_1', \ldots)$ и $B = \mathfrak{M}_2(E_0''; E_1'', \ldots)$ являются деревьями с корнями a и b, где $\mathfrak{M}_1 \cap \mathfrak{M}_2 = \Lambda$, $E_0' = (a)$ и $E_0'' = (b)$. Тогда сеть $C = \mathfrak{M}(E_0; E, E_1', \ldots)$ является деревом с корнем c, если

$$\mathfrak{M} = \mathfrak{M}_1 \cup \{c\}, \quad E_0 = (c), \quad E = (a, c),$$

где c — новый объект.

Далее, сеть $D = \mathfrak{M}'(E_0; \tilde{E}_1', \ldots, \tilde{E}_1'', \ldots)$ является деревом с корнем в вершине c, если

$$\mathfrak{M}' = (\mathfrak{M}_1 \backslash a) \cup (\mathfrak{M}_2 \backslash b) \cup \{c\}, \quad E_0 = (c)$$

и набор $\tilde{E}_i'$ (соответственно $\tilde{E}_i''$) получается из набора $E_i'(E_i'')$ заменой всех вхождений символа $a(b)$ на c, где c — новый объект.

Геометрическое определение дерева позволяет осуществить его геометрическую реализацию на плоскости.

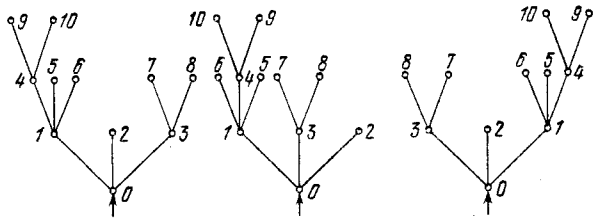


Рис. 7

Плоскую геометрическую реализацию дерева, в которой ребра представляют отрезки прямых, а корень изображен вершиной с дополнительным отрезком — стрелкой (см. рис. 6), будем называть *укладкой дерева*.

Пример 3. Пусть

$$\mathfrak{M} = \{0, 1, 2, \ldots, 10\}, \quad \mathfrak{N} = \{E_0; E_1, \ldots, E_{10}\},$$

где

$$E_0 = (0), \quad E_1 = (0, 1), \quad E_2 = (0, 2), \quad E_3 = (0, 3),$$
$$E_4 = (1, 4), \quad E_5 = (1, 5), \quad E_6 = (1, 6), \quad E_7 = (3, 7),$$
$$E_8 = (3, 8), \quad E_9 = (4, 9), \quad E_{10} = (4, 10).$$

Очевидно, что $\mathfrak{M}(E_0; E_1, \ldots, E_{10})$ — дерево. На рис. 7 изображено несколько укладок данного дерева.

Рассмотрим произвольную укладку дерева. Если двигаться по дереву от корня к концевым вершинам, то можно осуществить ориентацию ребер дерева. При этом в каждую вершину (включая корень) входит некоторое ребро и из каждой вершины, кроме концевых, исходит несколько ребер. Данное обстоятельство позволяет упорядочить исходящие ребра в каждой вершине, например, в порядке их следования, двигаясь от входящего ребра (см. рис. 8) по часовой стрелке.

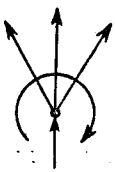


Рис. 8

Естественно считать, что две укладки одного дерева одинаковы, если порядки следования исходящих ребер для соответствующих вершин совпадают. Таким образом, укладка дерева полностью определяется порядками следования исходящих ребер.

§ 2. Оценка числа сетей

Мы начнем с рассмотрения простой задачи. Обозначим через $\delta(h)$ максимальное число попарно неизоморфных деревьев с h ребрами, а через $\delta^*(h)$ — число укладок деревьев из соответствующего множества.

Теорема 1. $\delta(h) \leqslant \delta^*(h) < 4^h$.

Доказательство. Так как укладки для неизоморфных деревьев различны, то $\delta(h) \leqslant \delta^*(h)$.

Каждой укладке дерева с h ребрами можно сопоставить взаимно однозначным образом кортеж из 0 и 1 длины $2h$. Для этого воспользуемся индуктивным определением дерева.

Базис индукции. Укладке дерева, содержащего ровно одно ребро, отнесем кортеж 01. Его длина равна 2.

Индуктивный переход. Пусть укладкам деревьев A и B, имеющих соответственно h_1 и h_2 ребер, сопоставлены кортежи α и β длины $2h_1$ и $2h_2$. Тогда укладке дерева C, полученной из укладки дерева A путем подключения ребра, сопоставим кортеж $0\alpha 1$. Его длина равна $2(h_1+1)$, т. е. удвоенному числу ребер дерева A. Далее, укладке дерева D, полученной из укладок деревьев A и B путем объединения корней, сопоставим $\alpha\beta$ или $\beta\alpha$ в зависимости от порядка их следования. Каждый из кортежей имеет длину $2(h_1+h_2)$, т. е. равную удвоенному числу ребер дерева D.

Заметим, что каждой укладке дерева соответствует кортеж, содержащий поровну нулей и единиц, причем в любом его начальном отрезке нулей не меньше, чем единиц. Если в каком-либо собственном начальном отрезке кортежа нулей и единиц поровну, то это означает, что данный кортеж соответствует укладке дерева D, полученной из укладок деревьев A и B путем объединения их корней. Отсюда видно, что по своему кортежу укладка дерева восстанавливается однозначно, т. е. имеется взаимно однозначное соответствие между укладками деревьев с h ребрами и подмножеством кортежей длины $2h$, содержащих поровну нулей и единиц.

Мы имеем

$$\delta^*(h) \leqslant \binom{2h}{h} < 2^{2h} = 4^h.$$

Теорема доказана.

Сравнивая верхние оценки для чисел $\gamma(h)$ и $\delta(h)$, связанных с числом графов и деревьев, мы видим, что последняя оценка существенно меньше, чем первая (при $h \to \infty$). Эта ситуация обусловлена тем, что деревья имеют более жесткую топологическую структуру.

Теперь мы перейдем к вопросу об оценке числа конечных сетей для общего случая. С этой целью введем ряд обозначений, касающихся сети

$$\mathfrak{M}(E_0;\ E_1,\ \ldots,\ E_h).$$

Пусть e_i обозначает число объектов (с учетом повторений) в наборе E_i. Величина e_i будет называться *степенью набора*. Обозначим через m максимальную степень набора, отличного от E_0, т. е.

$$m = \max_{1 \leqslant i \leqslant h} e_i.$$

Величину m будем называть *степенью сети*. Далее, пусть h_i $(1 \leqslant i \leqslant m)$ — число наборов степени i (без учета набора E_0). Кортеж $(h_1,\ h_2,\ \ldots,\ h_m)$ называется *степенной структурой сети*. Очевидно, $\sum_{i=1}^{m} h_i = h$. Наконец, введем величину

$$\mu = \frac{1}{h} \sum_{i=1}^{m} i h_i,$$

которую будем называть *средней степенью сети*.

Мы будем рассматривать класс сетей, для которых имеет место ограничение

$$\mathfrak{M} = \bigcup_{i=0}^{h} \langle E_i \rangle.$$

Это ограничение означает, что данные сети не имеют «изолированных» вершин, отличных от полюсных, и вершин, принадлежащих наборам.

Обозначим через $S(e_0, h_1, \ldots, h_m)$ максимальное число попарно неизоморфных сетей данного класса, имеющих e_0 полюсов и данную степенную структуру. Пусть $S(e_0, \mu, m, h)$ обозначает максимальное число попарно неизоморфных сетей того же класса, имеющих e_0 полюсов, данную среднюю степень μ, максимальную степень m и число наборов h (исключая полюсный набор).

Для указанных величин имеют место оценки, составляющие содержание теоремы 2 и ее следствия.

Теорема 2 (О. Б. Лупанов [17]).

$$S(e_0, h_1, \ldots, h_m) \leqslant c(e_0, \mu, m)^h h^{(\mu-1)h}.$$

Доказательство. Очевидно, что число вершин в наборах $E_1, \ldots, E_h$ не превосходит величины

$$p = \sum_{i=1}^{m} i h_i = \mu h.$$

Поскольку сети рассматриваются с точностью до изоморфизма, то можно считать, что полюсами являются

$$a_1, a_2, \ldots, a_{e_0}.$$

Произведем оценку числа p_i сортов наборов степени i, встречающихся в данных сетях. Поскольку среди этих сетей имеются и сети с p вершинами, для данной величины p_i имеют место соотношения

$$p_i = H_p^i = \binom{p+i-1}{i} \leqslant (p+i-1)^i \leqslant (2p)^i = (2\mu h)^i,$$

так как

$$i \leqslant m \leqslant \mu h = p.$$

Заметим, что в силу монотонности $\binom{p+i-1}{i}$ по i

$$p_i \geqslant \binom{p}{1} = p = \mu h \geqslant h_i.$$

Легко видеть, что число систем наборов степени i, каждая из которых содержит h_i наборов, не превосходит $H_{p_i}^{h_i}$. При $h_i \neq 0$ имеем

$$H_{p_i}^{h_i} = \binom{p_i + h_i - 1}{h_i} \leqslant \frac{(p_i + h_i - 1)^{h_i}}{h_i!} \leqslant \frac{(2p_i)^{h_i}}{(h_i/e)^{h_i}} = \frac{(2ep_i)^{h_i}}{h_i^{h_i}}.$$

Отсюда мы получаем оценку для величины $S(e_0, h_1, \ldots, h_m)$, которую обозначим просто через S:

$$S \leqslant (e_0 + 1) \prod_{\substack{i=1 \\ h_i \neq 0}}^{m} H_{p_i}^{h_i}.$$

Множитель $e_0 + 1$ отражает тот факт, что либо ни один из полюсов не принадлежит множеству $\bigcup_{i=1}^{h} \langle E_i \rangle$, либо один полюс принадлежит множеству $\bigcup_{i=1}^{h} \langle E_i \rangle$ и т. д., либо, наконец, все e_0 полюсов принадлежат $\bigcup_{i=1}^{h} \langle E_i \rangle$. Мы имеем

$$S \leqslant (e_0 + 1) \prod_{\substack{i=1 \\ h_i \neq 0}}^{m} \frac{(2e)^{h_i} (2\mu h)^{ih_i}}{h_i^{h_i}}$$

или

$$\ln S \leqslant \ln (e_0 + 1) + \sum_{\substack{i=1 \\ h_i \neq 0}}^{m} \ln \frac{(2e)^{h_i} (2\mu h)^{ih_i}}{h_i^{h_i}} =$$

$$= \ln (e_0 + 1) + h (\ln 2e + \mu \ln 2\mu) + \mu h \ln h - \sum_{\substack{i=1 \\ h_i \neq 0}}^{m} h_i \ln h_i.$$

Пусть $h_i = \xi_i h$. Очевидно, что $\sum_{i=1}^{m} \xi_i = 1$. При этих условиях

имеет место неравенство (для энтропии)*)

$$-\sum_{i=1}^{m} \xi_i \ln \xi_i \leqslant \ln m.$$

(При $\xi = 0$ положим $\xi \ln \xi = 0$.) Мы получаем

$$\ln S \leqslant \ln(e_0+1) + h(\ln m + \ln 2e + \mu \ln 2\mu) + (\mu - 1) h \ln h.$$

Отсюда имеем

$$S \leqslant (e_0 + 1)\,(2em\,(2\mu)^{\mu})^h h^{(\mu-1)h}.$$

Если положить $c(e_0, \mu, m) = (e_0 + 1)\,2em\,(2\mu)^{\mu}$, то окончательно получим

$$S(e_0, h_1, \ldots, h_m) \leqslant c(e_0, \mu, m)^h h^{(\mu-1)h}.$$

Теорема доказана.

Полученная оценка слабо зависит от степенной структуры $(h_1, \ldots, h_m)$: в нее входят лишь две ее характеристики, μ и m. Это позволяет легко получить и оценку для $S(e_0, \mu, m, h)$ при любых фиксированных значениях e_0, μ и m. Для этого надо оценить число степенных струк-

---

*) Из неравенства между средним геометрическим и средним арифметическим

$$(x_1 \ldots x_n)^{1/n} \leqslant \frac{1}{n}(x_1 + \ldots + x_n)$$

легко вывести следующее неравенство:

$$z_1^{\xi_1} \ldots z_m^{\xi_m} \leqslant \xi_1 z_1 + \ldots + \xi_m z_m, \qquad (*)$$

где все ξ_i $(i = 1, \ldots, m)$ — неотрицательные рациональные числа и

$$\sum_{i=1}^{m} \xi_i = 1.$$

Действительно, пусть n — общий знаменатель чисел $\xi_1, \ldots, \xi_m$ и $\xi_i = n_i/n$ $(i = 1, \ldots, m)$ (заметим, что $n_1 + n_2 + \ldots + n_m = n$). Тогда, применяя неравенство между средним геометрическим и средним арифметическим к n числам, среди которых первые n_1 чисел равны z_1, следующие n_2 чисел равны z_2 и т. д., мы и получим неравенство $(*)$.

Путем предельного перехода неравенство $(*)$ можно распространить и на иррациональные числа $\xi_1, \ldots, \xi_m$, однако в данном случае этого даже не требуется. Логарифмируя неравенство $(*)$, мы получим неравенство

$$\xi_1 \ln z_1 + \ldots + \xi_m \ln z_m \leqslant \ln(\xi_1 z_1 + \ldots + \xi_m z_m),$$

которое при $z_1 = 1/\xi_1, \ldots, z_m = 1/\xi_m$ превращается в неравенство для энтропии.

тур $(h_1, \ldots, h_m)$ с заданными параметрами μ, m, h. Последнее мажорируется числом целых неотрицательных решений уравнения $h_1 + h_2 + \ldots + h_m = h$. Это сводится к расстановке $m-1$ перегородок между h единицами. Мы удовлетворимся грубой оценкой $(h+1)^{m-1}$ (каждая перегородка может занимать $h+1$ положение). Таким образом, получаем

Следствие.

$$S(e_0, \mu, m, h) \leqslant c(e_0, \mu, m)^h (h+1)^{m-1} h^{(\mu-1)h} \leqslant$$
$$\leqslant c'(e_0, \mu, m)^h h^{(\mu-1)h}.$$

Полученная оценка как частный случай содержит и оценку для числа графов

$$\gamma(h) = S(0, 2, 2, h) \leqslant c^h h^h.$$

Отсюда получается оценка и для числа графов с двумя выделенными вершинами и без изолированных неполюсных вершин (эту оценку легко получить и из следствия теоремы 3 гл. 1)

$$S(2, 2, 2, h) \leqslant c_3^h h^h.$$

§ 3. Двухполюсные сети из двухобъектных наборов

Здесь мы рассматриваем важный класс конечных сетей, имеющих два различных полюса ($e_0 = 2$) и состоящих исключительно из двухобъектных наборов ($e_i = 2$, при $i = 1, 2, \ldots, h$). Легко видеть, что данный класс совпадает с классом конечных графов, в каждом из которых выделены две вершины — полюса. Подобного рода сети

$$\mathfrak{M}(E_0; E_1, \ldots, E_h)$$

мы будем обозначать через $\Gamma(a, b)$, где $E_0 = (a, b)$.

Для сетей, так же как и для графов, вводится понятие пути, соединяющего некоторые его вершины a^0, b^0. В случае, если вершины a^0 и b^0 совпадают с полюсами a и b соответственно, то мы употребляем термин «путь» без указания вершин, которые он соединяет. Сеть называется *связной*, если соответствующий граф связен. Если сеть $\Gamma(a, b)$ связна, то для каждого ребра можно указать путь, проходящий через него. Заметим, что для связных

сетей

$$\mathfrak{M} \subseteq \bigcup_{i=1}^{h} \langle E_i \rangle.$$

Следовательно, связная сеть полностью определяется перечислением ее ребер и указанием полюсов.

Дальнейшие рассуждения этого параграфа будут относиться исключительно к связным сетям.

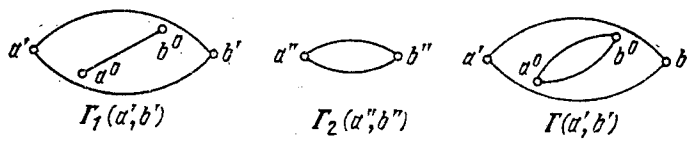


Рис. 9

Пусть $\Gamma_1(a', b')$ и $\Gamma_2(a'', b'')$ — две непересекающиеся связные сети, т. е.

$$\Gamma_1(a', b') = \mathfrak{M}_1(E'_0; E'_1, \ldots, E'_{h'}),$$
$$\Gamma_2(a'', b'') = \mathfrak{M}_2(E''_0; E''_1, \ldots, E''_{h''}),$$

где $\mathfrak{M}_1 \cap \mathfrak{M}_2 = \Lambda$. Рассмотрим произвольное ребро $E'_i = (a^0, b^0)$ сети $\Gamma_1(a', b')$. Исходя из геометрических соображений (см. рис. 9), нетрудно дать определение операции подстановки вместо ребра E'_i сети $\Gamma_2(a'', b'')$, приводящей к новой сети $\Gamma(a', b')$.

Определение. *Результатом подстановки вместо ребра* $E'_i = (a^0, b^0)$, *принадлежащего сети* $\Gamma_1(a', b')$, *сети* $\Gamma_2(a'', b'')$ называется каждая из сетей $\Gamma'(a', b')$ и $\Gamma''(a', b')$:

$$\Gamma'(a', b') = \mathfrak{M}(E'_0; E'_1, \ldots, E'_{i-1}, E^{\mathrm{III}}_1, \ldots, E^{\mathrm{III}}_{h''}, E'_{i+1}, \ldots, E'_{h'}),$$
$$\Gamma''(a', b') = \mathfrak{M}(E'_0; E'_1, \ldots, E'_{i-1}, E^{\mathrm{IV}}_1, \ldots, E^{\mathrm{IV}}_{h''}, E'_{i+1}, \ldots, E'_{h'}),$$

где $\mathfrak{M} = \mathfrak{M}_1 \cup (\mathfrak{M}_2 \setminus \{(a'', b'')\})$.

Набор E^{III}_j $(j = 1, \ldots, h'')$ получается из набора E''_j заменой a'' на a^0 и b'' на b^0, набор E^{IV}_j $(j = 1, \ldots, h'')$ получается из набора E''_j заменой a'' на b^0 и b'' на a^0.

Определение. Сеть $\Gamma(a, b)$, получающаяся из сетей, изоморфных сетям

$$\Gamma_1(a', b'), \ldots, \Gamma_m(a^{(m)}, b^{(m)}),$$

путем применения конечного числа операции подстановки, называется *суперпозицией этих сетей.*

Пример 4. Сеть $\Gamma(a, b)$, изображенная на рис. 10, является суперпозицией сетей $\Gamma_1(a', b')$, $\Gamma_2(a'', b'')$, $\Gamma_3(a''', b''')$ (см. рис. 11).

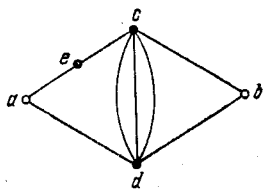


Рис. 10

В самом деле, возьмем сеть $\Gamma_4(a^{IV}, b^{IV})$, изоморфную сети $\Gamma_3(a''', b''')$, и подставим ее вместо ребра сети $\Gamma_3(a''', b''')$. Полученную сеть подставим в сеть $\Gamma_1(a', b')$ вместо ребра (c, d). Затем, осуществляя подстановку в этой промежуточной сети вместо ребра (a', c) сети $\Gamma_2(a'', b'')$, получим сеть $\Gamma(a, b)$.

Замечания. 1. Легко видеть, что операция суперпозиции является ассоциативной операцией.

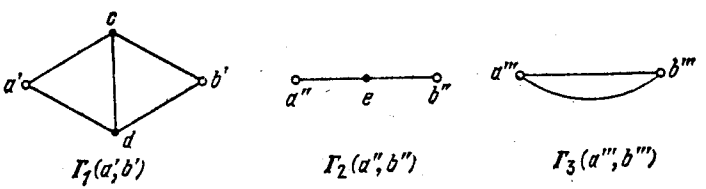


Рис. 11

2. Множество всех связных сетей $\{\Gamma(a, b)\}$ вместе с операцией суперпозиции определяет функциональную систему с операцией.

Пусть в сети $\Gamma(a, b)$ взяты два пути $A'_{a^0b^0}$ и $A''_{a^0b^0}$, соединяющие вершины a^0 и b^0.

Определение. Путь

$$A''_{a^0b^0} = \{(a^0, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{s-1}}, b^0)\}$$

называется *подпутем* пути

$$A'_{a^0b^0} = \{(a^0, a_{j_2}), (a_{j_2}, a_{j_3}), \ldots, (a_{j_{t-1}}, b^0)\},$$

если последовательность ребер

$$\{(a^0, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{s-1}}, b^0)\}$$

получается из последовательности ребер

$$\{(a^0, a_{j_2}), (a_{j_2}, a_{j_3}), \ldots, (a_{j_{t-1}}, b^0)\}$$

путем удаления некоторого подмножества ребер. Подпуть $A''_{a^0b^0}$ пути $A'_{a^0b^0}$, отличный от самого пути $A'_{a^0b^0}$, называется *собственным подпутем*.

Определение. Путь $A_{a^0b^0}$, соединяющий вершины a^0 и b^0 сети $\Gamma(a, b)$, называется *цепью, соединяющей эти вершины*, если он не содержит собственных подпутей.

Замечание. В случае, если вершины a^0 и b^0 совпадают с полюсами a и b, вместо слов «цепь, соединяющая a и b», будем говорить просто «цепь».

Очевидно, что путь является цепью тогда и только тогда, когда он не проходит дважды через одну вершину.

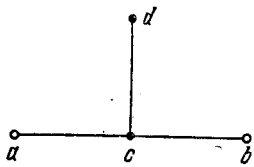


Рис. 12

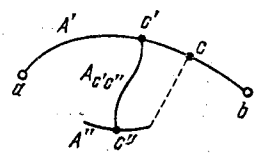


Рис. 13

Пример 5. Рассмотрим сеть $\Gamma(a, b)$, изображенную на рис. 12. Очевидно, что $\{(a, c), (c, d), (d, c), (c, b)\}$ является путем, но не является цепью, так как содержит собственный подпуть $\{(a, c), (c, b)\}$. Путь $\{(a, c), (c, b)\}$ является цепью.

Легко видеть, что сеть, содержащая h $(h > 0)$ ребер, имеет бесконечное число путей и конечное число цепей.

Введем понятие, которое позволит еще сузить класс рассматриваемых сетей.

Определение. Связная сеть $\Gamma(a, b)$ называется *сильно связной*, если через каждое ее ребро проходит некоторая цепь.

Очевидно, что не всякая связная сеть является сильно связной (см. пример 5).

Ниже доказываются две леммы *). Первая из них дает условие, при котором через данное ребро можно провести цепь. Она служит основой для доказательства второй лем-

*) Дальше изложение связано с работами А. В. Кузнецова [16] и Б. А. Трахтенброта [31].

мы, выясняющей необходимые и достаточные условия сильной связности.

Лемма 1. *Пусть* $\Gamma(a, b)$ — *произвольная сеть* (*не обязательно связная*) *и пусть через вершины* c' *и* c'' ($c' \neq c''$) *сети* Γ *проходят цепи* A' *и* A'' (*не исключено, что* $A' = A''$). *Если вершины* c' *и* c'' *можно соединить цепью* $A_{c'c''}$, *имеющей с цепями* A' *и* A'' *общими только концевые вершины* c' *и* c'', *то существует цепь* A, *частью которой является* $A_{c'c''}$.

Доказательство. Если обе вершины c' и c'' принадлежат одновременно хотя бы одной цепи, например, A', то тогда искомая цепь A получается из A' заменой части цепи A', расположенной между c' и c'', на $A_{c'c''}$. В противном случае вершины c' и c'' являются внутренними. Обозначим через c первую общую для цепей A' и A'' вершину, если двигаться по цепи A'' от вершины c'' к полюсу b ($c \neq c'$ и $c \neq c''$). На рис. 13 изображена одна из двух возможных ситуаций. Обозначим через A путь, получающийся из цепи A' заменой участка $c'c$ на путь, состоящий из цепи $A_{c'c''}$ и участка цепи A'' между вершинами c'' и c. Очевидно, A является искомой цепью.

Пусть в сети $\Gamma(a, b)$ выделено некоторое подмножество ребер Γ', которое, очевидно, определяет граф. Вершина c графа Γ' называется *граничной*, если она является либо полюсом сети $\Gamma(a, b)$, либо концом ребра сети $\Gamma(a, b)$, не принадлежащего Γ'.

Определение. Подмножество Γ' ребер сети $\Gamma(a, b)$ называется *отростком*, если Γ' обладает единственной граничной вершиной.

Например, на рис. 12 подмножество ребер $\{(c, d)\}$ является отростком, так как имеет одну граничную вершину c, а подмножество ребер $\{(a, c), (c, b)\}$ отростком не является, поскольку оно имеет три граничные вершины: a, c, b.

Лемма 2. *Связная сеть* $\Gamma(a, b)$ *является сильно связной тогда и только тогда, когда* $\Gamma(a, b)$ *не содержит отростков.*

Доказательство. Пусть связная сеть $\Gamma(a, b)$ содержит отросток Γ'. Обозначим через c его граничную вершину. Рассмотрим произвольное ребро, принадлежащее этому отростку. Ясно, что всякий путь, проходящий через данное ребро, должен по крайней мере два раза пройти через вершину c. Ввиду этого через ребро не про-

ходит ни одной цепи. Следовательно, сеть $\Gamma(a, b)$ не является сильно связной.

Пусть теперь связная сеть $\Gamma(a, b)$ не является сильно связной. Покажем, что тогда она содержит отросток. Так как $\Gamma(a, b)$ не является сильно связной, то существуют ребра, через которые не проходит ни одной цепи. Пусть Γ' — максимальное связное подмножество ребер, обладающих этим свойством*). В силу связности сети $\Gamma(a, b)$ граф Γ' обладает по крайней мере одной граничной вершиной. Предположим, что Γ' имеет по крайней мере две граничные вершины. Так как граф Γ' связен, то каждая пара граничных вершин может быть соединена цепью, целиком принадлежащей Γ'. Пусть c' и c'' — две такие граничные вершины, что указанная цепь $A_{c'c''}$ не содержит никаких других граничных вершин. Ясно, что через вершины c' и c'' можно провести цепи (соединяющие полюса) A' и A''. Очевидно, что цепь $A_{c'c''}$ имеет с цепями A' и A'' общими только концевые вершины c' и c''. Применяя к цепям A', A'' и $A_{c'c''}$ лемму 1, мы построим цепь, частью которой будет $A_{c'c''}$, что противоречит определению Γ'. Следовательно, Γ' обладает единственной граничной вершиной, т. е. Γ' является отростком. Лемма доказана.

Рис. 14

В дальнейшем будем рассматривать только сильно связные сети. Рассмотрим сеть $\Gamma_0(a, b) = \mathfrak{M}_0(E_0; E_1)$, где $\mathfrak{M} = \{a, b\}$, $E_0 = E_1 = (a, b)$ (см. рис. 14). Эту сеть будем называть *тривиальной сетью*.

Определение. Сильно связная сеть называется *разложимой*, если существуют такие нетривиальные непересекающиеся сети $\Gamma_1(a, b)$ и $\Gamma_2(a', b')$, что сеть $\Gamma(a, b)$ есть результат подстановки сети $\Gamma_2(a', b')$ вместо некоторого ребра сети $\Gamma_1(a, b)$. Очевидно, сеть, приведенная на рис. 10, является разложимой.

Определение. Сильно связная сеть $\Gamma(a, b)$, не являющаяся разложимой, называется *неразложимой*.

На рис. 15 представлены примеры трех нетривиальных неразложимых сетей (полюса помечены кружочками).

Можно показать, что любая нетривиальная неразложимая сильно связная сеть, имеющая не более шести ребер, изоморфна одной из трех сетей, представленных на рис. 15. Это означает, что не существует неразложимых

*) Если их несколько, возьмем одно из них.

сильно связных сетей с тремя, четырьмя и шестью ребрами. В то же время для каждого $h \geqslant 7$ существуют неразложимые сильно связные сети с h ребрами (см. [31]). Последнее легко усматривается из рис. 16, на котором

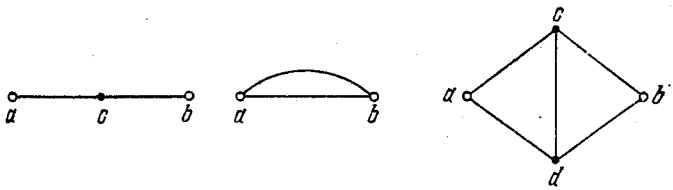


Рис. 15

указано построение неразложимых сетей с h $(h \geqslant 7)$ ребрами.

Цель дальнейших рассмотрений — изучение вопросов разложимости сетей. Поскольку нас будут интересовать разложения специального вида, выделим две простейшие

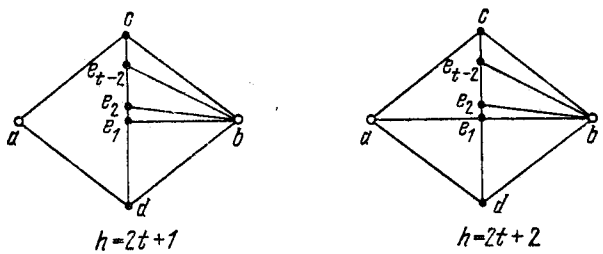


Рис. 16

неразложимые сети: параллельное соединение двух ребер $\Gamma_2^p(a, b)$ и последовательное соединение двух ребер $\Gamma_2^s(a, b)$ (см. рис. 17). Множество остальных нетривиальных неразложимых сетей обозначим через H и сеть, принадлежащую H, будем называть *H-сетью*. С сетями $\Gamma_2^p(a, b)$ и $\Gamma_2^s(a, b)$ связаны два бесконечных множества сетей.

Множество P, состоящее из сетей $\Gamma_k^p(a, b)$, — параллельное соединение k ребер $(k = 2, 3, \ldots)$. Очевидно, что $\Gamma_k^p(a, b)$ при $k > 2$ является суперпозицией сетей $\Gamma_2^p(a, b)$ (см. рис. 18).

Множество S, состоящее из сетей $\Gamma_k^s(a, b)$, — последовательное соединение k ребер ($k = 2, 3, \ldots$). Очевидно, что $\Gamma_k^s(a, b)$ при $k > 2$ является суперпозицией сетей $\Gamma_2^s(a, b)$ (см. рис. 18).

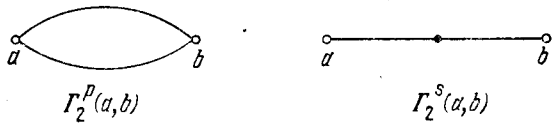


Рис. 17

Определение. Сильно связная сеть $\Gamma(a, b)$ *распадается на два параллельных куска*, если множество всех ее цепей можно разбить на два непустых класса так, что любые две цепи из разных классов не имеют общих внутренних вершин. Очевидно, каждая из сетей $\Gamma_k^p(a, b)$ ($k \geqslant 2$) распадается на два параллельных куска.

Рис. 18

Определение. Пусть c — внутренняя (т. е. отличная от полюсов) вершина сильно связной сети $\Gamma(a, b)$. Вершина c называется *разделяющей*, если каждая цепь проходит через c. Очевидно, что каждая из внутренних вершин сети $\Gamma_k^s(a, b)$ является разделяющей.

Пусть c — разделяющая вершина сети $\Gamma(a, b)$. Рассмотрим в каждой цепи отрезок от вершины a до вершины c. Очевидно, что совокупность этих отрезков цепей порождает сеть $\Gamma_1(a, c)$ с полюсами в вершинах a и c. Аналогично, если выделить в каждой цепи отрезок от вершины c до вершины b, то получим сеть $\Gamma_2(c, b)$.

Лемма 3. *Пусть c — разделяющая вершина сильно связной сети $\Gamma(a, b)$. Тогда $\Gamma(a, b)$ получается суперпозицией сетей $\Gamma_2^s(a, b)$, $\Gamma_1(a, c)$, $\Gamma_2(c, b)$ (последовательное соединение сетей $\Gamma_1(a, c)$ и $\Gamma_2(c, b)$).*

Доказательство следует из того факта, что сети $\Gamma_1(a, c)$ и $\Gamma_2(c, b)$ не имеют общих внутренних вершин. В самом деле, если это не так, то существуют две цепи A_{ac} и A_{cb}, соответственно, сетей $\Gamma_1(a, c)$ и $\Gamma_2(c, b)$, которые имеют общую внутреннюю вершину (см. рис. 19).

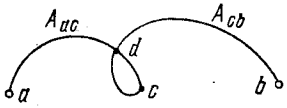


Рис. 19

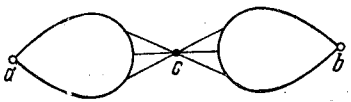


Рис. 20

Обозначим через d первую общую вершину этих цепей, если двигаться по цепи A_{ac} от точки a. Очевидно, что отрезок цепи A_{ac} между вершинами a и d и отрезок цепи A_{cb} между вершинами d и b порождают цепь, не проходящую через c. Последнее противоречит тому, что вершина c является разделяющей. Лемма доказана.

Определение. Пусть c — вершина сети $\Gamma(a, b)$, тогда совокупность всех ее ребер, имеющих своим концом вершину c, называется *звездой* (*с центром в* c). Если c — полюс, то звезда называется *полюсной*.

Относительно H-сетей сформулируем следующую лемму.

Лемма 4. *Если* $\Gamma(a, b)$ — *H-сеть, а* c *и* d — *две различные внутренние вершины* (*т. е. отличные от полюсов*), *то существует цепь, проходящая через* d *и не проходящая через вершину* c.

Доказательство. Данное утверждение вытекает из более сильного факта: если из $\Gamma(a, b)$ удалить звезду с центром в c, то полученная сеть будет сильно связной.

Возможны три случая.

а) Удаление звезды приводит к распадению графа на две связные компоненты, которые не имеют общих вершин (см. рис. 20). Очевидно, в этом случае вершина c будет разделяющей, и в силу леммы 3 исходная сеть будет разложимой (одна из компонент будет содержать не менее двух ребер, так как H-сеть имеет более четырех ребер), а это невозможно.

б) Удаление звезды приводит к сети, которая является связной, но не сильно связной. В силу того, что полученная сеть не сильно связная, она имеет отросток с граничной точкой d (см. рис. 21). Поскольку исходная сеть

$\Gamma(a, b)$ сильно связная, то отросток должен иметь граничные вершины с данной звездой и отличные от d. В таком случае отросток с частью ребер звезды образует двухполюсную сеть $\Gamma'(d, c)$. Последнее означает, что $\Gamma(a, b)$ — разложима, мы пришли к противоречию.

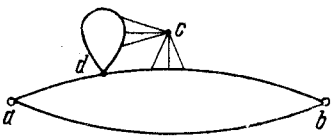


Рис. 21

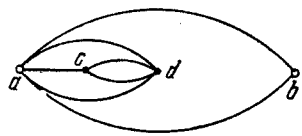


Рис. 22

в) Остается последняя возможность — удаление звезды приводит к сильно связной сети. Лемма доказана.

Следствие. *H-сеть не имеет разделяющих вершин.*

Лемма 5. *Если* $\Gamma(a, b)$ — *H-сеть и* (a, c) — *ребро, принадлежащее полюсной звезде, то после удаления этого ребра получим сильно связную сеть.*

Доказательство (аналогично доказательству предыдущей леммы).

а) Удаление ребра дает сеть, не являющуюся связной. Очевидно, тогда c будет разделяющей вершиной сети $\Gamma(a, b)$, что противоречит следствию леммы 4.

б) Удаление ребра дает связную сеть $\Gamma'(a, b)$, но не сильно связную. Обозначим граничную вершину отростка сети $\Gamma'(a, b)$ через d (см. рис. 22). Ясно, что этот отросток вместе с ребром (a, c) дает двухполюсную сеть $\Gamma''(a, d)$. Последнее противоречит неразложимости $\Gamma(a, b)$.

в) Остается последняя возможность: сеть $\Gamma'(a, b)$ — сильно связная.

Рассмотрим разложимую сеть $\Gamma(a, b)$. Пусть $\Gamma(a, b)$ есть результат подстановки вместо ребер $E_1, \ldots, E_h$ $(h \geqslant 2)$ сети $\Gamma_0(a, b) = \mathfrak{M}(E_0; E_1, \ldots, E_h)$ соответственно сетей $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$, из которых хотя бы одна нетривиальна.

Разложение сети $\Gamma(a, b)$ на внешнюю сеть $\Gamma_0(a, b)$ и внутренние сети $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ допускает простое геометрическое толкование: исходная сеть $\Gamma(a, b)$ покрывается сетями $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ так, что любые две внутренние сети могут иметь общими только свои полюсные вершины; расположение этих внутренних

сетей характеризуется внешней сетью (см. рис. 23). Таким образом, каждое ребро сети принадлежит ровно одной внутренней сети, а вершина сети $\Gamma(a, b)$ либо является полюсом внутренней сети (и значит вершиной внешней

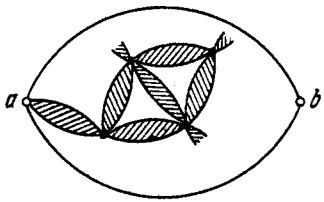


Рис. 23

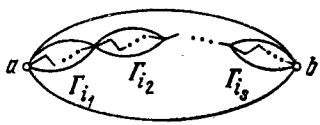


Рис. 24

сети), либо внутренней вершиной ровно одной внутренней сети.

Замечание. Выберем в сети $\Gamma_0(a, b)$ цепь. Пусть

$$\Gamma_{i_1}(a^{(i_1)}, b^{(i_1)}), \ldots, \Gamma_{i_s}(a^{(i_s)}, b^{(i_s)})$$

— внутренние сети, соответствующие ребрам этой цепи. Если в этих сетях выбрать по одной цепи, то получим цепь в $\Gamma(a, b)$ (см. рис. 24).

Определение. Пусть сеть $\Gamma(a, b)$ разлагается на внешнюю сеть $\Gamma_0(a, b)$ и внутренние сети $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$. Тогда:

а) если $\Gamma_0(a, b)$ есть $\Gamma_h^p(a, b)$, то сеть $\Gamma(a, b)$ называется *p-разложимой*, а соответствующее разложение — *p-разложением*;

б) если $\Gamma_0(a, b)$ есть $\Gamma_h^s(a, b)$, то сеть $\Gamma(a, b)$ называется *s-разложимой*, а соответствующее разложение — *s-разложением*;

в) если $\Gamma_0(a, b)$ — H-сеть, то сеть $\Gamma(a, b)$ называется *H-разложимой*, а соответствующее разложение — *H-разложением*.

Лемма 6. *Если сеть* $\Gamma(a, b)$ *разложима, то имеет место в точности одно из следующих утверждений*:

$\Gamma(a, b)$ — *p-разложима*,

$\Gamma(a, b)$ — *s-разложима*,

$\Gamma(a, b)$ — *H-разложима*.

Доказательство. Нетрудно показать, что, если $\Gamma(a, b)$ разложима, то она допускает либо p-разложение, либо s-разложение, либо H-разложение. А именно, так

как $\Gamma(a, b)$ разложима, то она допускает разложение на $\Gamma_0(a, b)$ (внешняя сеть) и $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ (внутренние сети). Если внешняя сеть есть либо $\Gamma_h^p(a, b)$, либо $\Gamma_h^s(a, b)$, либо сеть класса H, то мы имеем одно из указанных разложений. Если это не так, то $\Gamma_0(a, b)$ разложима и мы получаем другое разложение сети $\Gamma(a, b)$ на сеть $\Gamma_0'(a, b)$ (внешняя сеть) и $\Gamma_1'(a_1^{(1)}, b_1^{(1)}), \ldots$ $\ldots, \Gamma_{h'}'(a_1^{(h')}, b_1^{(h')})$ (внутренние сети), причем сеть $\Gamma_0'(a, b)$ имеет меньше ребер, чем сеть $\Gamma_0(a, b)$ $(h' < h)$. В случае, если внешняя сеть $\Gamma_0'(a, b)$ есть либо $\Gamma_{h'}^p(a, b)$, либо $\Gamma_{h'}^s(a, b)$, либо сеть класса H, мы получаем искомое разложение. В противном случае сеть $\Gamma_0'(a, b)$ разложима и процесс повторяем снова. Так как сеть $\Gamma(a, b)$ имеет конечное число ребер, то в конце концов мы построим требуемое разложение. Теперь остается показать, что тип разложения определяется единственным образом.

Предположим, что сеть $\Gamma(a, b)$ допускает H-разложение.

а) Тогда она не может распадаться на два параллельных куска. Иначе внешняя сеть распадается также на два куска, т. е. является разложимой либо совпадает с $\Gamma_2^p(a, b)$, что при H-разложении невозможно.

б) Она не может иметь и разделяющей вершины. В противном случае разделяющей вершиной будет внутренняя вершина внешней сети. Последнее при H-разложении невозможно в силу следствия леммы 4.

Ясно также, что два свойства: распадение сети на два параллельных куска и наличие разделяющей вершины — взаимоисключающи. Таким образом, разложимая двухполюсная сеть обладает в точности одним из следующих свойств: распадается на два параллельных куска, имеет разделяющую вершину, допускает H-разложение.

I. Разложимая сеть распадается на два параллельных куска в том и только том случае, если она p-разложима.

II. Разложимая сеть имеет разделяющую вершину тогда и только тогда, когда она s-разложима.

III. Разложимая сеть не распадается на два параллельных куска и не имеет разделяющей вершины в том и только в том случае, если она H-разложима.

Лемма доказана.

Определение. p-разложение сети $\Gamma(a, b)$ называется *p-расщеплением*, если внутренние сети разложения

отличны от сетей вида $\Gamma_2^p(a, b)$ и сетей, являющихся p-разложимыми; s-разложение сети $\Gamma(a, b)$ называется *s-расщеплением*, если внутренние сети разложения отличны от сетей вида $\Gamma_2^s(a, b)$ и сетей, являющихся s-разложимыми; H-разложение сети $\Gamma(a, b)$ называется *H-расщеплением*.

Заметим, что сети Γ_k^p и Γ_k^s $(k \geqslant 3)$ являются разложимыми, но не допускающими расщепления.

Пример 6. Рассмотрим сеть, изображенную на рис. 25. Очевидно, что разложение этой сети на сеть $\Gamma_0'(a, b)$ и сети $\Gamma_1'(a, c)$, $\Gamma_2'(c, b)$ (см. рис. 26, *а*)) не является s-расщеплением, так как сеть $\Gamma_2'(c, b)$ есть сеть вида $\Gamma_2^s(c, b)$. В то же время s-разложение на сеть $\Gamma_0''(a, b)$ и сети $\Gamma_1''(a, c)$, $\Gamma_2''(c, d)$, $\Gamma_3''(d, b)$ (см.рис. 26, *б*)) является s-расщеплением.

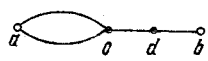


Рис. 25

Данный пример показывает, что для сети $\Gamma(a, b)$ может существовать несколько разложений.

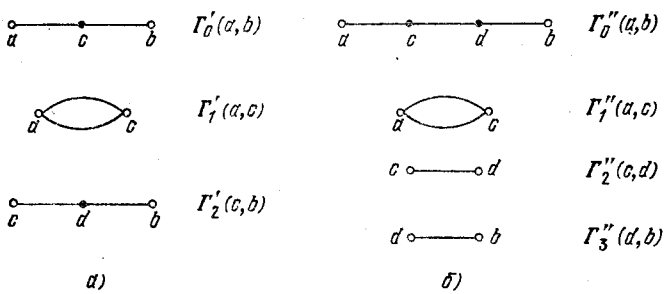


Рис. 26

Определение. Внутренняя вершина c сети $\Gamma(a, b)$ *зависит от вершины* d той же сети, если каждая цепь, проходящая через c, проходит также через d.

Определение. Внутренние вершины c и d сети $\Gamma(a, b)$ называются *эквивалентными*, если вершина c зависит от вершины d и вершина d зависит от вершины c.

Определение. Внутренняя вершина c сети $\Gamma(a, b)$ называется *минимальной*, если, какова бы ни была внутренняя вершина d сети $\Gamma(a, b)$, либо c эквивалентна d, либо c не зависит от d.

Пример 7. Рассмотрим сеть $\Gamma(a, b)$, изображенную на рис. 27. Здесь вершина f зависит от вершины e, вершина e эквивалентна вершине g, вершины c и d являются минимальными.

Из определений следует, что отношения зависимости и эквивалентности удовлетворяют аксиоме транзитивности, а отношение эквивалентности — также и рефлексивности.

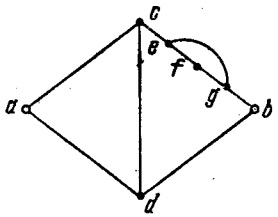


Рис. 27

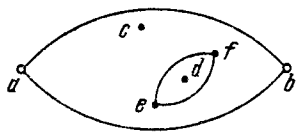


Рис. 28

Лемма 7. *Если* $\Gamma(a, b)$ *допускает* H*-разложение, то совокупность минимальных вершин совпадает с совокупностью внутренних вершин внешней сети.*

Доказательство. Пусть c — минимальная вершина; покажем, что она принадлежит множеству внутренних вершин внешней сети. Допустим, что это не так. Тогда c принадлежит некоторой внутренней сети. Очевидно, что c зависит от полюсных вершин этой сети. Так как внешняя сеть — H-сеть, то по крайней мере одна из данных полюсных вершин является внутренней вершиной исходной сети. В этом случае c не является минимальной вершиной.

Пусть теперь c — внутренняя вершина внешней сети, покажем, что она минимальна. Для этого достаточно установить, что c не зависит ни от какой другой внутренней вершины d, т. е. что существует цепь, проходящая через c и не проходящая через d. Очевидно, вершина d либо является внутренней вершиной внешней сети, и тогда мы воспользуемся леммой 4, либо является внутренней вершиной некоторой внутренней сети (см. рис. 28).

Обозначим через e и f полюса этой сети. Возможны следующие подслучаи.

а) Вершина e совпадает с полюсом, например a, вершина f совпадает с вершиной c. Тогда применяем лемму 5. Учитывая замечание на с. 247, мы получим цепь, проходящую через c и не проходящую через d.

б) Одна из вершин, например f, не совпадает ни с полюсом сети ни с вершиной c. Здесь применим лемму 4. Учитывая замечание на с. 247, мы опять получим цепь, проходящую через c и не проходящую через d.

Теорема 3. *Если сильно связная сеть* $\Gamma(a, b)$ *разложима и отлична от сетей* Γ_h^p, $\Gamma_k^i (k \geqslant 3)$, *то она допускает единственное расщепление.*

Доказательство. Мы уже видели, что тип расщепления единствен. Поэтому остается показать единственность расщепления внутри данного типа.

Случай 1. Сеть $\Gamma(a, b)$ распадается на два параллельных куска. Проведем разбиение всех цепей сети на классы. Две цепи A' и A'' относятся к одному классу тогда и только тогда, когда на цепях A' и A'' найдутся внутренние вершины (соответственно c' и c''), которые можно соединить цепью $A_{c'c''}$, не проходящей через полюса (отсюда следует, что и любая пара внутренних вершин этих цепей может быть соединена цепью, не проходящей через полюса). Такое определение корректно, поскольку если внутренние вершины цепей A' и A'' могут быть соединены цепями, не проходящими через полюса, и внутренние вершины цепей A'' и A''' — цепями, не проходящими через полюса, то и внутренние вершины цепей A' и A''' могут быть соединены цепями, не проходящими через полюса. Таким образом, мы получаем разбиение на классы

$$\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_h,$$

и это разбиение определяется однозначным образом. Данное разбиение порождает p-расщепление на сети

$$\Gamma_h^p(a, b), \Gamma_1(a, b), \ldots, \Gamma_h(a, b)$$

(каждая из сетей $\Gamma_i(a, b)$ $(i = 1, \ldots, h)$ отлична от $\Gamma_2^p(a, b)$ и не допускает p-разложений; кроме того, хотя бы одна из них нетривиальна, так как $\Gamma(a, b) \neq \Gamma_h^p(a, b)$).

Случай 2. Сеть $\Gamma(a, b)$ имеет разделяющие вершины $c_1, \ldots, c_{h-1}$. Легко видеть, что, двигаясь по любой цепи от полюса a к полюсу b, разделяющие вершины встречаются в одном и том же порядке (пусть $c_1, \ldots, c_{h-1}$). Пусть $\Gamma_1(a, c_1)$, $\Gamma_2(c_1, c_2)$, $\ldots$, $\Gamma_h(c_{h-1}, b)$ — сети, образованные из отрезков этих цепей между соответствующими вершинами. Как и в доказательстве леммы 3, легко показать, что эти сети не имеют общих вершин, отличных от своих

полюсов. Мы приходим к s-расщеплению сети $\Gamma(a, b)$ на сеть $\Gamma_h^s(a, b)$ и сети $\Gamma_1(a_1, c_1)$, $\Gamma_2(c_1, c_2)$, ..., $\Gamma_h(c_{h-1}, b)$, так как $\Gamma(a, b) \neq \Gamma_h^s(a, b)$, и, значит, хотя бы одна из внутренних сетей нетривиальна. Из построения вытекает однозначность.

Случай 3. Сеть $\Gamma(a, b)$ допускает H-разложение. Как это следует из леммы 7, оно однозначно.

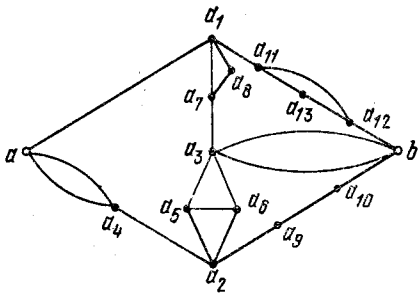


Рис. 29

В самом деле, минимальные вершины задают все внутренние вершины внешней сети, пара минимальных вершин или полюсов c, d задает ребро внешней сети тогда и только тогда, когда существует цепь, соединяющая c и d и не проходящая через другие минимальные вершины или полюса. Теорема доказана.

Рассмотрим некоторую сеть $\Gamma(a, b)$. Если она допускает расщепление, то осуществим ее расщепление. В результате получим сети с меньшим числом ребер. Опять

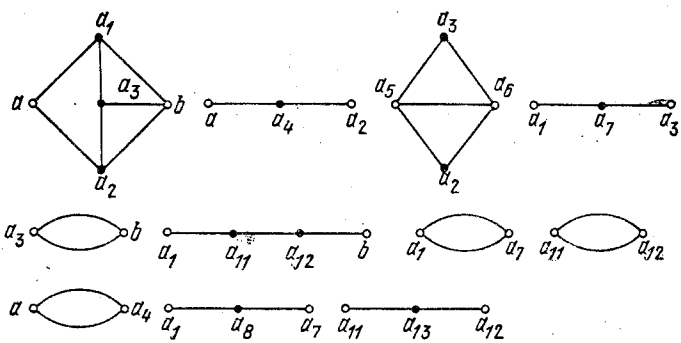


Рис. 30

расщепим те внутренние сети, которые допускают расщепление, и т. д. Этот процесс закончится на конечном шаге и мы придем к сетям либо тривиальным, либо неразложимым, либо сетям вида Γ_k^p, Γ_k^s $(k \geqslant 3)$. Система всех нетривиальных сетей, которая возникает при рас-

щеплении сети $\Gamma(a, b)$, называется *каноническим расщеплением сети* $\Gamma(a, b)$. Таким образом, нами доказана

Теорема 4. *Каждая сильно связная нетривиальная сеть* $\Gamma(a, b)$, *допускающая расщепление, имеет единственное каноническое расщепление.*

Пример 8. Сеть $\Gamma(a, b)$, изображенная на рис. 29, при каноническом расщеплении разлагается на сети (см. рис. 30).

§ 4. π-сети

Важным подклассом двухполюсных сетей из двухобъектных наборов является класс π-сетей.

Определение. Сеть, являющаяся суперпозицией сетей $\Gamma_2^p(a, b)$ и $\Gamma_2^s(a, b)$, называется *π-сетью.*

Данное определение эквивалентно другому: двухполюсная сеть из двухобъектных наборов, которая сильно связна, будет π-сетью, если каноническое расщепление содержит сети двух типов $\Gamma_h^p(a, b)$ и $\Gamma_h^s(a, b)$.

Пример 9. Сеть $\Gamma(a, b)$, изображенная на рис. 31, будет π-сетью.

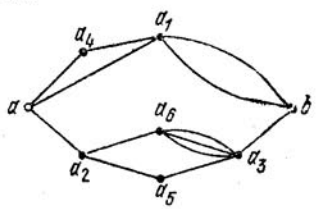


Рис. 31

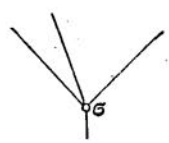


Рис. 32

Каждой π-сети можно поставить в соответствие множество укладок дерева, неконцевым вершинам которых сопоставлены символы p и s. Возможны два случая:

1) $\Gamma(a, b) = \Gamma_h^\sigma(a, b)$, где $\sigma = p$ или $\sigma = s$. Сети $\Gamma_h^\sigma(a, b)$ ставим в соответствие пучок из h ($h \geqslant 2$) ребер *), корень которого помечен символом σ (рис. 32).

2) $\Gamma(a, b)$ расщепляется на сеть $\Gamma_h^\sigma(a, b)$ и сети $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ ($\sigma = p$ или $\sigma = s$). Выпускаем из корня, помеченного символом σ, h ($h \geqslant 2$) ребер *),

*) Порядок ребер в пучке при $\sigma = p$ произволен, при $\sigma = s$ соответствует порядку ребер в сети $\Gamma_h^s(a, b)$.

которые соответствуют внутренним сетям (рис. 32). Далее, концам ребер, которым соответствуют нетривиальные сети, приписываем символ, отличный от σ (его обозначим через $\bar{\sigma}$*)). После этого для каждой нетривиальной сети

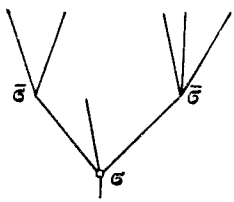


Рис. 33

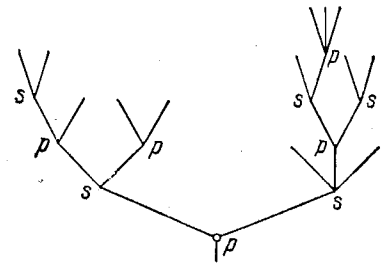


Рис. 34

$\Gamma_i(a^{(i)}, b^{(i)})$ применяют либо п. 1, либо п. 2 и строят пучки в вершинах $\bar{\sigma}$ и т. д. (рис. 33). В построенной укладке дерева каждый пучок ребер содержит не менее двух ребер. Таким образом, каждой π-сети соответствует множество укладок дерева. Неизоморфным π-сетям соответствуют непересекающиеся множества укладок. Значит число π-сетей не превосходит числа укладок деревьев.

Рассмотрим укладку дерева для π-сети из примера 9 (рис. 34). Легко видеть, что укладка дерева, соответствующая π-сети с h ребрами, имеет h концевых вершин.

Итак, изучение π-сетей может быть сведено к изучению укладок деревьев специального вида. Покажем, что эта связь позволяет переносить некоторые факты, известные для деревьев, на π-сети.

Теорема 5. *Пусть* $\pi(h)$ *— максимальное число попарно неизоморфных* π*-сетей с* h *ребрами. Тогда* $\pi(h) \leqslant 4^{2h}$.

Доказательство. Очевидно, что искомое число не превосходит числа укладок выше указанного типа деревьев с h концевыми вершинами, у которых каждый пучок исходящих ребер содержит не менее двух ребер. Обозначим через l число ребер в дереве из этого класса. По индукции докажем, что $l \leqslant 2h - 2$ при $h \geqslant 2$.

Базис индукции. Если π-сеть содержит два ребра: $h = 2$, то очевидно, что соответствующее дерево содержит

*) Где $\bar{\sigma} = s$, если $\sigma = p$ и $\bar{\sigma} = p$, если $\sigma = s$.

две концевые вершины, т. е. $l=2$ и неравенство выполнено.

Индуктивный переход. Пусть неравенство верно для деревьев, соответствующих π-сетям с менее, чем h ребрами. Рассмотрим π-сеть $\Gamma(a, b)$ с h ребрами и соответствующее ей дерево (рис. 35). Если $\Gamma(a, b)=\Gamma_h^\sigma(a, b)$, то $l=h>2$ и неравенство, очевидно, выполнено. Если $\Gamma(a, b)$ допускает расщепление, то в дереве число m исходящих из корня ребер равно числу ребер внешней сети расщепления сети $\Gamma(a, b)$ и по условию $m\geqslant 2$. Деревья $D_1, \ldots, D_t$ соответствуют нетривиальным внутренним сетям этого расщепления, $t\leqslant m$. Обозначим через l_i и h_i $(i=1, \ldots, t)$ число ребер и число концевых вершин в дереве D_i. По предположению индукции $l_i\leqslant 2h_i-2$ $(i=1, \ldots, t)$, кроме того, очевидно, $\sum\limits_{i=1}^{t} l_i+m=l$, $\sum\limits_{i=1}^{t} h_i+(m-t)=h$. Мы имеем

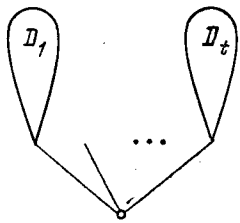


Рис. 35

$$l=\sum_{i=1}^{t} l_i+m\leqslant 2\sum_{i=1}^{t} h_i-2t+m=$$

$$=2\left(\sum_{i=1}^{t} h_i+(m-t)\right)-m=2h-m\leqslant 2h-2,$$

так как $m\geqslant 2$.

Для оценки величины $\pi(h)$, заметим, что в каждой укладке дерева из данного класса символы p и s можно расставить двумя способами. В силу этого, используя оценку для числа укладок деревьев с заданным числом ребер, имеем

$$\pi(h)\leqslant 2\cdot 4^{2h-2}<4^{2h}.$$

ЧАСТЬ IV
ТЕОРИЯ КОДИРОВАНИЯ

Вопросы кодирования играют существенную роль в математике. Кодирование позволяет изучение одних объектов сводить к изучению других. Хорошо известно, какую роль сыграло изображение чисел в десятичной системе счисления. Весьма важным в развитии математики было появление метода координат, который позволил

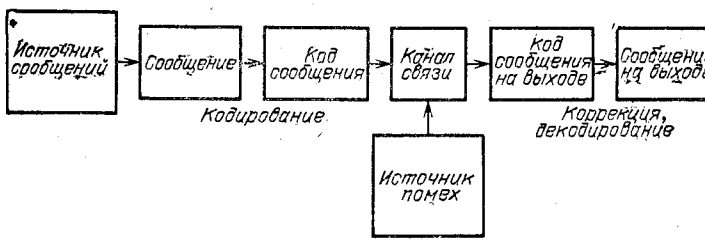


Рис. 1

кодировать геометрические объекты при помощи аналитических выражений. Однако, здесь средства кодирования являлись вспомогательным аппаратом и не были предметом изучения. Совсем другое значение получили коды в связи с изучением управляющих систем. Появилась необходимость систематического исследования в области теории кодирования. Основной круг задач может быть прослежен на примере из области связи, который характеризуется схемой, представленной на рис. 1.

Пусть задан алфавит $\mathfrak{A} = \{a_1, \ldots, a_r\}$, состоящий из конечного числа букв. Конечную последовательность символов из $\mathfrak{A}$

$$A = a_{i_1} a_{i_2} \ldots a_{i_n}$$

будем называть *словом* в алфавите $\mathfrak{A}$, а число n — *длиной* слова A. Длину слова A будем обозначать через $l(A)$.

Пусть $S = S(\mathfrak{A})$ — множество всех непустых слов в алфавите $\mathfrak{A}$, и S' — некоторое подмножество множества S. Объект, порождающий слова из S', называется *источником сообщений*, а слова из S' — *сообщениями*. Источником сообщений может быть автомат, человек и т. п. Обычно при рассмотрении задач теории кодирования используется дополнительная информация об источнике сообщений в виде некоторого его описания. Существует ряд способов описаний источников сообщений:

а) *теоретико-множественное описание* осуществляется путем фиксации мощностных характеристик S', например, S' — множество всех слов заданной длины m;

б) *статистическое описание* осуществляется заданием вероятностных характеристик S', например, $S' = S$, и заданы вероятности $p_1, \ldots, p_r$ появления букв $a_1, \ldots, a_r \left(\sum_{i=1}^{r} p_i = 1\right)$;

в) *логическое описание* — это описание множества S' как некоторого «языка». Оно характеризует способы построения множества S', например, S' может быть порождено некоторым автоматом.

Пусть задан алфавит $\mathfrak{B}$, где

$$\mathfrak{B} = \{b_1, \ldots, b_q\}.$$

Через B обозначим слово в алфавите $\mathfrak{B}$ и через $S(\mathfrak{B})$ — множество всех непустых слов в алфавите $\mathfrak{B}$.

Пусть задано отображение F, которое каждому слову A, $A \in S'(\mathfrak{A})$, ставит в соответствие слово

$$B = F(A), \quad B \in S(\mathfrak{B}).$$

Слово B будем называть *кодом сообщения* A, а переход от слова A к его коду — *кодированием*.

В теории кодирования отображения F задается некоторым алгоритмом.

Пример 1. а) *Алфавитное кодирование.* Рассмотрим соответствие между буквами алфавита $\mathfrak{A}$ и некоторыми словами в алфавите $\mathfrak{B}$:

$$\begin{aligned} a_1 &- B_1, \\ a_2 &- B_2, \\ &\ldots \\ a_r &- B_r. \end{aligned} \qquad (\Sigma)$$

Это соответствие называют *схемой* и обозначают через Σ. Оно определяет алфавитное кодирование следующим образом: каждому слову $A = a_{i_1} \dots a_{i_n}$ из $S'(\mathfrak{A}) = S(\mathfrak{A})$ ставится в соответствие слово $B = B_{i_1} \dots B_{i_n}$, называемое *кодом слова* A. Слова $B_1, \dots, B_r$ называются *элементарными кодами*.

б) *Равномерное кодирование.* Пусть $\{A_1, \dots, A_s\}$ — некоторое подмножество попарно различных слов в алфавите $\mathfrak{A}$, имеющих одинаковую длину m. Очевидно, что каждое слово A, которое допускает разложение вида

$$A = A_{i_1} \dots A_{i_n},$$

имеет единственное разложение. Пусть, далее, $S'(\mathfrak{A})$ — подмножество всех слов в алфавите $\mathfrak{A}$, допускающих разложение вышеуказанного вида. Рассмотрим схему

$$\begin{aligned} A_1 &- B_1, \\ &\dots \\ A_s &- B_s, \end{aligned} \qquad (\Sigma)$$

где элементарные коды B_i имеют одинаковую длину.

Схема Σ определяет равномерное кодирование следующим образом: каждому слову $A = A_{i_1} \dots A_{i_n}$ из $S'(\mathfrak{A})$ ставится в соответствие слово $B = B_{i_1} \dots B_{i_n}$, называемое *кодом слова* A.

Выбор кодов связан с различными обстоятельствами, а именно:

с удобством передачи кодов (например, двоичный код технически легче использовать);

с обеспечением удобства восприятия (например, машинные коды удобны для работы процессора);

с обеспечением максимальной пропускной способности канала;

с обеспечением помехоустойчивости;

с достижением определенных свойств алгоритма кодирования (например, простота кодирования, возможность однозначного декодирования) и т. п.

Канал связи можно рассматривать как устройство с одним входом и одним выходом (см. рис. 2). На вход этого устройства поступает код сообщения B. На выходе получают B' — код сообщения на выходе, где B' — слово в некотором алфавите $\mathfrak{B}'$ и

$$B' = f(B).$$

В простейшем случае (тождественный канал без помех), т. е. в случае идеальной линии связи, $B' \equiv B$ (или $f(B) = B$) и значит $\mathfrak{B}' = \mathfrak{B}$. В общем случае канал связи может включать в себя преобразование кодов и $\mathfrak{B}' \neq \mathfrak{B}$ (как, например, в ЭВМ).

Источник помех вносит ошибки в канал связи, вызывая искажения кодов на выходе. Для описания источника помех используют два способа:

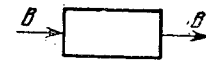


Рис. 2

а) *логико-комбинаторное описание* обычно связано с указанием ограничений на число единичных ошибок;

б) *статистическое описание* осуществляется заданием вероятностных характеристик источника.

Код сообщения на выходе в случае тождественного канала представляет собой некоторое слово в алфавите $\mathfrak{B}' = \mathfrak{B}$. Однако источник помех может приводить к тому, что

$$B' \neq B.$$

Сообщение на выходе представляет собой слово в некотором алфавите $\mathfrak{C}$. В случае тождественного канала, т. е. при передаче сообщений, $\mathfrak{C} = \mathfrak{A}$.

Переход от кодов сообщений на выходе к сообщениям на выходе предполагает два преобразования.

Коррекция кода сообщения на выходе. Это преобразование возможно только для специальных кодов сообщений. В том случае, если мы имеем дело с передачей сообщений, происходит переход от B' к B.

Декодирование. Оно представляет собой переход от кода, полученного из кода сообщения на выходе после коррекции, к сообщению на выходе. Декодирование возможно также не для всяких кодов, а только для специальных кодов сообщений. В случае передачи сообщений декодирование возможно, если существует обратное отображение F^{-1}.

В данной главе мы познакомимся с элементами теории кодирования. При отборе материала мы стремились дать представление:

а) о главных классах кодов;

б) о теоретико-вероятностных и комбинаторно-логических подходах к описанию задач;

в) о характере математических задач в этой области;

г) о методах решения задач теории кодирования.

В § 1—4 изучается алфавитное кодирование. Изложение концентрируется вокруг двух задач: выяснения возможности однозначного декодирования и построения кодов с наименьшей избыточностью. В § 5 изучается один класс кодов из семейства так называемых равномерных кодов. Здесь рассматривается задача о построении помехоустойчивых кодов.

§ 1. Критерий однозначности декодирования

Здесь мы рассматриваем алфавитное кодирование для алфавитов $\mathfrak{A}$ и $\mathfrak{B}$, задаваемое схемой Σ:

$$\begin{aligned} a_1 &- B_1, \\ &\cdots \\ a_r &- B_r, \end{aligned} \qquad (\Sigma)$$

и полагаем $S'(\mathfrak{A}) = S(\mathfrak{A})$, т. е. источник сообщений порождает множество всех слов в алфавите $\mathfrak{A}$. Очевидно, что алфавитное кодирование порождает отображение множества $S(\mathfrak{A})$ в множество $S(\mathfrak{B})$. Обозначим через $S_\Sigma(\mathfrak{B})$ образ множества $S(\mathfrak{A})$ в этом отображении.

В случае, когда отображение $S(\mathfrak{A})$ на $S_\Sigma(\mathfrak{B})$ взаимно однозначно, возможно декодирование, т. е. возможно по коду B однозначно восстановить исходное сообщение A, кодом которого является B. В этом случае говорят также, что алфавитное кодирование является взаимно однозначным.

Пример 2. Рассмотрим алфавитное кодирование, для которого $\mathfrak{A} = \{a_1, a_2\}$, $\mathfrak{B} = \{b_1, b_2\}$ и схема имеет вид

$$\begin{aligned} a_1 &- b_1, \\ a_2 &- b_1 b_2. \end{aligned}$$

Пусть B' и B'' являются соответственно кодами слов A' и A''. Очевидно, что если $A' \neq A''$, то $B' \neq B''$.

Процесс декодирования происходит следующим образом. Произведем разбиение слова B, $B \in S_\Sigma(\mathfrak{B})$, на элементарные коды. Для этого заметим, что перед каждым вхождением буквы b_2 в слове B непосредственно находится буква b_1. Это позволяет выделить все пары $(b_1 b_2)$. Оставшаяся часть слова B будет состоять из букв b_1. Если теперь заменить каждую пару $(b_1 b_2)$ на a_2, а каждую из оставшихся букв b_1 — на a_1, то получим слово A, являющееся прообразом B.

Пусть $B = b_1b_1b_2b_1b_2b_1b_1b_1b_2$. После выделения пар мы получим разбиение на элементарные коды

$$B = b_1(b_1b_2)\,(b_1b_2)\,b_1b_1(b_1b_2),$$

из которого находим слово

$$A = a_1a_2a_2a_1a_1a_2.$$

Можно привести большое количество примеров, в которых алфавитное кодирование не будет обладать свойством взаимной однозначности. В связи с этим возникает вопрос: возможно ли по схеме Σ алфавитного кодирования узнать, обладает алфавитное кодирование свойством взаимной однозначности или нет. Трудность решения задачи состоит в том, что для непосредственной проверки взаимной однозначности необходимо просмотреть бесконечное число слов.

Прежде чем приводить общий критерий взаимной однозначности алфавитного кодирования, рассмотрим весьма простой достаточный признак взаимной однозначности.

Определение. Пусть слово B имеет вид

$$B = B'B''.$$

Тогда слово B' называется *началом* или *префиксом слова* B, а B'' — *концом слова* B. При этом пустое слово Λ и само слово B считаются началами и концами слова B. Все начала и концы слова B, отличные от него самого, называются *собственными*.

Определение. Схема Σ *обладает свойством префикса*, если для любых i и j $(1 \leqslant i,\ j \leqslant r,\ i \neq j)$ слово B_i не является префиксом слова B_j.

Теорема 1. *Если схема* Σ *обладает свойством префикса, то алфавитное кодирование будет взаимно однозначным.*

Доказательство. Поскольку схема Σ обладает свойством префикса, все элементарные коды в ней попарно различны, т. е. $B_i \neq B_j$ при $i \neq j$. Предположим, что некоторое слово B из $S_\Sigma(\mathfrak{B})$ допускает две расшифровки, а значит и два разбиения на элементарные коды

$$B = B_{i_1} \ldots B_{i_s},$$

$$B = B_{j_1} \ldots B_{j_t}.$$

Пусть $B_{i_1} = B_{j_1}, \ldots, B_{i_{n-1}} = B_{j_{n-1}}, B_{i_n} \neq B_{j_n}$. В таком случае одно из слов B_{i_n}, B_{j_n} является префиксом другого. Теорема доказана.

Предыдущий пример показывает, что условие префиксности не является необходимым: Σ может не обладать свойством префикса, а алфавитное кодирование, определяемое Σ, будет взаимно однозначным.

Пусть $B = b_{i_1} \ldots b_{i_n}$ — слово из $S(\mathfrak{B})$. Обозначим через $\widetilde{B}$ — слово, получающееся из B путем «обращения», т. е.

$$\widetilde{B} = b_{i_n} \ldots b_{i_1}.$$

Обозначим, далее, через $\widetilde{\Sigma}$ схему вида

$$a_1 - \widetilde{B}_1,$$

$$\ldots$$

$$a_r - \widetilde{B}_r.$$

Пример 3. Возьмем в качестве Σ схему из примера 2. Тогда $\widetilde{\Sigma}$ имеет вид

$$a_1 - b_1,$$

$$a_2 - b_2 b_1.$$

Здесь $\widetilde{\Sigma}$ обладает свойством префикса, и в силу теоремы 1 алфавитное кодирование, задаваемое $\widetilde{\Sigma}$, будет взаимно однозначным.

Замечание. Алфавитное кодирование, определяемое схемой Σ, и алфавитное кодирование, определяемое схемой $\widetilde{\Sigma}$, одновременно либо обладают свойством взаимной однозначности, либо нет.

Данное замечание позволяет усилить теорему 1.

Теорема 2. *Если либо схема* Σ, *либо схема* $\widetilde{\Sigma}$ *обладает свойством префикса, то алфавитное кодирование, задаваемое* $\Sigma(\widetilde{\Sigma})$, *будет взаимно однозначным.*

Можно привести пример алфавитного кодирования со схемой Σ так, что и Σ и $\widetilde{\Sigma}$ не обладают свойством префикса, а алфавитное кодирование будет взаимно однозначным.

Для этого предыдущий пример уже не годится, но может быть легко усовершенствован.

Пример 4. Пусть $\mathfrak{A} = \{a_1, a_2, a_3\}$ и $\mathfrak{B} = \{b_1, b_2, b_3\}$. Рассмотрим схему Σ:

$$\begin{aligned} a_1 &- b_1, \\ a_2 &- b_1 b_2, \\ a_3 &- b_3 b_1. \end{aligned} \qquad (\Sigma)$$

Очевидно, Σ и $\tilde{\Sigma}$ не обладают свойством префикса, в то же время алфавитное кодирование будет взаимно однозначным. В самом деле, если $B \in S_\Sigma(\mathfrak{B})$, то это слово однозначным образом разбивается на элементарные коды:

слева от буквы b_2 непосредственно находится b_1 — выделяем пару $(b_1 b_2)$;

справа от буквы b_3 непосредственно находится b_1 — выделяем пару $(b_3 b_1)$;

после выделения всех пар $(b_1 b_2)$ и $(b_3 b_1)$ в слове останутся лишь символы b_1.

В дальнейшем предположим, что в Σ элементарные коды попарно различны.

Прежде чем идти дальше, введем ряд обозначений: $l(B)$ будет обозначать длину слова B, т. е. количество букв в этом слове. В частности для длин элементарных кодов B_i $(i = 1, \ldots, r)$ полагаем $l(B_i) = l_i$. Далее, через L обозначим величину $l(B_1 \ldots B_r)$, т. е. «длину» схемы Σ.

Пусть

$$B_i = \beta' B_{i_1} \ldots B_{i_w} \beta'' \qquad (1)$$

— нетривиальное разложение элементарного кода B_i, т. е. разложение, отличное от разложения

$$B_i = B_i \qquad (\beta' = \beta'' = \Lambda),$$

причем β' и β'' отличны от элементарных кодов.

Параметр w может быть целым числом, большим или равным нулю.

Соотношение (1) означает, что в элементарном коде B_i можно отбросить некое начало β' и некий конец β'' так, что оставшаяся часть разбивается на элементарные коды (см. рис. 3).

Очевидно, что для каждого B_i число разложений вида (1) конечно. Обозначим через W максимум чисел w, взятый по всем разложениям B_i и по всем i, т. е.

$$W = \max w.$$

Пример 5. Рассмотрим алфавитное кодирование с $\mathfrak{A} = \{a_1, a_2, a_3, a_4, a_5\}$, $\mathfrak{B} = \{b_1, b_2, b_3\}$ и схемой

$$a_1 - b_1b_2,$$
$$a_2 - b_1b_3b_2,$$
$$a_3 - b_2b_3,$$
$$a_4 - b_1b_2b_1b_3,$$
$$a_5 - b_2b_1b_2b_2b_3.$$

Так как $6 > l_i \geqslant 2$, то $W < 3$. С другой стороны,

$$B_5 = b_2b_1b_2b_2b_3 = b_2B_1B_3,$$

поэтому $W = 2$.

Обозначим, наконец, через $S^N(\mathfrak{A})$ множество всех непустых слов в алфавите $\mathfrak{A}$, имеющих длину, не превосходящую N. Ясно, что $S^N(\mathfrak{A})$ — конечное множество, его мощность равна $\sum_{i=1}^{N} r^i$.

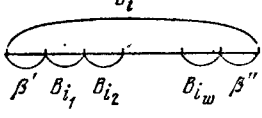


Рис. 3

Сформулируем теперь критерий взаимной однозначности алфавитного кодирования.

Теорема 3 (Марков Ал. А. [22, 23]). *Для всякого алфавитного кодирования со схемой* Σ *существует такое* N_0,

$$N_0 \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right],$$

что проблема взаимной однозначности алфавитного кодирования сводится к аналогичной проблеме для кодирования конечного множества $S^{N_0}(\mathfrak{A})$.

Доказательство. Если в алфавитном кодировании нарушена взаимная однозначность, то найдется такое слово B, которое допускает по крайней мере две различные расшифровки A' и A''. Для доказательства теоремы достаточно установить, что можно найти также такое слово B, что для его расшифровок A' и A'' имеют место неравенства

$$l(A'),\ l(A'') \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

Слово B, допускающее не менее двух расшифровок, называется *неприводимым*, если каждое слово B', получающееся из B путем выбрасывания непустого куска, допускает не более одной расшифровки.

Ясно, что с самого начала можно предполагать, что слово B — неприводимо. Рассмотрим его две расшифров-

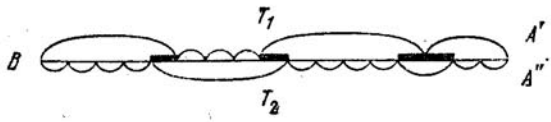


Рис. 4

ки A' и A''. Очевидно, что с ними связаны два разбиения слова B на элементарные коды — верхнее T_1 и нижнее T_2 (см. рис. 4).

Возьмем произведение T этих разбиений, т. е. разбиение, которое получается путем одновременного разбиения T_1 и T_2. Слова этого нового разбиения T разделим на два класса: к первому классу отнесем слова, являющиеся элементарными кодами, ко второму — все остальные слова. Ясно, что слова из первого класса входят каждое ровно в одно разбиение, а слова второго класса (на рис. 4 изображены жирными отрезками) являются одновременно непустыми началами и концами элементарных кодов из разных разбиений и отличны от элементарных кодов, так как слово B неприводимо. Покажем, что каждые два слова β' и β'' из второго класса различны, т. е. $\beta' \neq \beta''$. Положим противное: $\beta' = \beta'' = \beta$. Тогда

$$B = B'\beta'B''\beta''B''' = B'\beta B''\beta B'''.$$

Утверждается, что слово $B'\beta'B'''$, получающееся из B путем выбрасывания куска $B''\beta''$, допускает по крайней мере две расшифровки. Для расположения слов β' и β'' возможны 4 случая, которые изображены на рис. 5. Нетрудно видеть, что случай *в*) сводится к случаю *а*), случай *г*) — к случаю *б*). Для этого в *в*) и *г*) нужно поменять ролями разбиения T_1 и T_2.

В случае *а*) при выбрасывании куска $B''\beta''$ и сдвигании кусков $B'\beta'$ и B''' верхнее разбиение получается из склейки частей верхнего разбиения, нижнее — из склейки частей нижнего разбиения.

В случае *б*) верхнее разбиение получается из склейки части верхнего разбиения для куска $B'\beta'$ и нижнего разбиения для B''', нижнее разбиение — путем склейки нижнего разбиения для куска $B'\beta'$ и верхнего — для куска B'''.

Итак, для слова $B'\beta'B'''$ во всех случаях мы получаем по крайней мере две расшифровки, что противоречит тому, что исходное слово неприводимо.

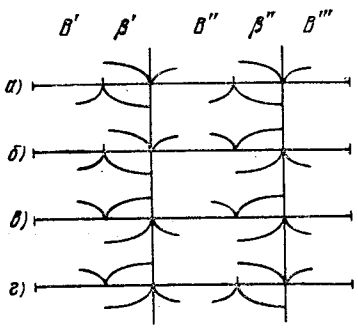


Рис. 5

Число p слов второго класса не превосходит числа непустых собственных начал элементарных кодов, т. е.

$$p \leqslant (l(B_1)-1)+ \\ +(l(B_2)-1)+\ldots \\ \ldots+(l(B_r)-1)=L-r.$$

Слова из второго класса разбивают слово B не более чем на $L-r+1$ кусков, некоторые из них могут быть пустыми (см. рис. 6).

Пусть $\beta^0=\beta^{p+1}=\Lambda$. Рассмотрим один из кусков, расположенный между словами β^j и β^{j+1} $(j=0, \ldots, p)$:

$$\beta^j B_{i_1}\ldots B_{i_w}\beta^{j+1}.$$

В этом куске все слова $B_{i_1}, \ldots, B_{i_w}$ принадлежат одному и тому же разбиению слова B, например T_1, а слово

$$\beta^j B_{i_1}\ldots B_{i_w}\beta^{j+1}=B_i$$

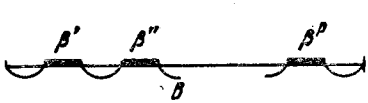


Рис. 6

принадлежит другому разбиению — T_2. При этом $w \leqslant W$. Поэтому с каждым куском связаны w элементарных кодов одного разбиения и один элементарный код другого разбиения.

Если взять теперь два соседних куска

$$\beta^j B_{i_1'}\ldots B_{i_{w'}'}\beta^{j+1}B_{i_1''}\ldots B_{i_{w''}''}\beta^{j+2},$$

то элементарные коды

$$B_{i_1'}, \ldots, B_{i_{w'}'}, \quad B_{i''}=\beta^{j+1}B_{i_1''}\ldots B_{i_{w''}''}\beta^{j+2}$$

входят в одно разбиение, а

$$B_{i_1''}, \ldots, B_{i_{w''}''}, \quad B_{i'} = \beta^j B_{i_1'}' \ldots B_{i_{w'}'}' \beta^{j+1}$$

— в другое разбиение (см. рис. 7).

Следовательно, кускам разбиения T между словами β^j и β^{j+1} для одной и той же четности j соответствуют

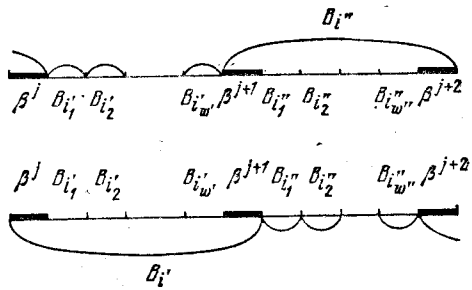


Рис. 7

элементарные коды B_i исходного разбиения (например T_1) и группы элементарных кодов $B_{i_1}, \ldots, B_{i_w}$ того же разбиения для другой четности j. Отсюда легко получается оценка для максимального числа элементарных кодов, входящих в разбиение. Оно не превосходит

$$\left[\frac{L-r+2}{2}\right] \cdot 1 + \left[\frac{L-r+2}{2}\right] W \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

Мы получаем, что

$$l(A'), l(A'') \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

Если теперь положить $N_0 = \max(l(A'), l(A''))$, то очевидно, что взаимная однозначность кодирования уже нарушается на множестве $S^{N_0}(\mathfrak{A})$, так как $A', A'' \in \in S^{N_0}(\mathfrak{A})$. Теорема полностью доказана.

Критерий однозначности декодирования дает простой алгоритм для установления по схеме Σ, будет алфавитное кодирование обладать свойством взаимной однозначности или нет. Для этого достаточно рассмотреть множество слов в алфавите $\mathfrak{A}$, имеющих длину не более N_0, т. е. $S^{N_0}(\mathfrak{A})$, и выяснить, будет ли кодирование этого ко-

нечного множества взаимно однозначным. Трудоемкость такого алгоритма можно грубо оценить как r^{N_0}. Оказывается, что данный алгоритм не может быть использован даже в простых примерах.

Пример 6. Рассмотрим алфавитное кодирование (см. пример 5). Мы имеем $r=5$, $W=2$, $L=16$. Берем $N_0=\left[\frac{(W+1)(L-r+2)}{2}\right]=\left[\frac{3\cdot 13}{2}\right]=19$ и получаем

$$r^{N_0}=5^{19},$$

что весьма велико.

§ 2. Алгоритм распознавания однозначности декодирования

Из доказательства критерия однозначности декодирования можно извлечь достаточно эффективный алгоритм для распознавания возможности декодирования. Этот алгоритм формулируется на языке теории графов (Марков Ал. А. [22—24]).

Пусть алфавитное кодирование задано схемой Σ:

$$\begin{aligned} &a_1 - B_1,\\ &\cdot\ \cdot\ \cdot\ \cdot\ \cdot\\ &a_r - B_r. \end{aligned} \qquad (\Sigma)$$

Для каждого элементарного кода B_i рассмотрим все нетривиальные представления вида

$$B_i=\beta' B_{i_1}\dots B_{i_w}\beta'', \qquad (1)$$

в которых β' и β'' отличны от элементарных кодов. Обозначим через $\mathfrak{B}_0$ множество, содержащее:

а) пустое слово Λ;

б) слова β, которые встречаются в разложениях вида (1) как в форме префиксов, так и окончаний.

Далее, каждому слову из $\mathfrak{B}_0$ сопоставим точку на плоскости.

Пусть $\beta', \beta''\in\mathfrak{B}_0$. Рассмотрим все нетривиальные разложения вида (1). Для каждого из них соединяем соответствующие словам β' и β'' вершины направленным отрезком (от β' к β''), которому приписано слово $B_{i_1}\dots B_{i_w}$. Полученный граф обозначим через $\Gamma(\Sigma)$.

Отметим особо тривиальный случай, когда свойство взаимной однозначности алфавитного кодирования со схемой Σ нарушается из-за того, что существует элементарный код B_i с разложением вида (1), в котором $\beta' = \beta'' = \Lambda$ (т. е. B_i разбивается на элементарные коды). В этом случае граф $\Gamma(\Sigma)$ содержит петлю при вершине Λ.

Теорема 4. *Алфавитное кодирование со схемой* Σ *не обладает свойством взаимной однозначности тогда и только тогда, когда* $\Gamma(\Sigma)$ *содержит ориентированный цикл, проходящий через вершину* Λ.

Доказательство. Необходимость. Пусть алфавитное кодирование не обладает свойством взаимной однозначности. Тогда найдется неприводимое слово B вида (см. § 1)

$$B = B_{i_1^0} \dots B_{i_{w0}^0} \beta' B_{i_1^1} \dots B_{i_{w1}^1} \beta'' \dots \beta^p B_{i_1^p} \dots B_{i_{wp}^p},$$

для которого

$$B_{i^0} = B_{i_1^0} \dots B_{i_{w0}^0} \beta',$$

$$B_{i^1} = \beta' B_{i_1^1} \dots B_{i_{w1}^1} \beta'',$$

$$\dots\dots\dots\dots$$

$$B_{i^p} = \beta^p B_{i_1^p} \dots B_{i_{wp}^p}.$$

Поэтому в графе $\Gamma(\Sigma)$ имеется ориентированный цикл (см. рис. 8), проходящий через вершину Λ.

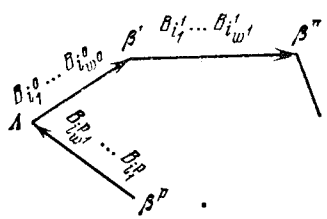


Рис. 8

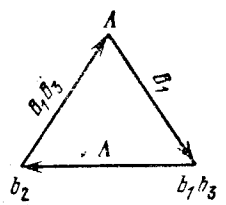


Рис. 9

Достаточность. Пусть $\Gamma(\Sigma)$ содержит ориентированный цикл, проходящий через вершину Λ (см.

рис. 8). Тогда слово B, где

$$B = B_{i_1^0} \dots B_{i_{w^0}^0} \beta' B_{i_1^1} \dots B_{i_{w^1}^1} \beta'' \dots \beta^p B_{i_1^p} \dots B_{i_{w^p}^p},$$

имеет две расшифровки, определяемые двумя разбиениями

$$B = \left(B_{i_1^0}\right) \dots \left(B_{i_{w^0}^0}\right) \left(\beta' B_{i_1^1} \dots B_{i_{w^1}^1} \beta''\right) \left(B_{i_1^2}\right) \dots$$
$$\dots \left(B_{i_{w^2}^2}\right) \left(\beta''' B_{i_1^3} \dots B_{i_{w^3}^3} \beta^{IV}\right) \dots,$$
$$B = \left(B_{i_1^0} \dots B_{i_{w^0}^0} \beta'\right) \left(B_{i_1^1}\right) \dots \left(B_{i_{w^1}^1}\right) \left(\beta'' B_{i_1^2} \dots B_{i_{w^2}^2} \beta'''\right) \dots$$

Теорема доказана.

Таким образом, алгоритм состоит из построения графа $\Gamma(\Sigma)$ и выявления ориентированных циклов, проходящих через Λ.

Пример 7. Рассмотрим алфавитное кодирование (см. пример 5) со схемой

$$\begin{aligned} a_1 &- b_1 b_2, \\ a_2 &- b_1 b_3 b_2, \\ a_3 &- b_2 b_3, \\ a_4 &- b_1 b_2 b_1 b_3, \\ a_5 &- b_2 b_1 b_2 b_2 b_3. \end{aligned} \qquad (\Sigma)$$

Имеем следующие нетривиальные разложения:

$$\begin{aligned} B_1 &= (b_1)(b_2), \\ B_2 &= (b_1)(b_3 b_2) = (b_1 b_3)(b_2), \\ B_3 &= (b_2)(b_3), \\ B_4 &= (b_1)(b_2 b_1 b_3) = (b_1 b_2)(b_1 b_3) = (b_1 b_2 b_1)(b_3), \\ B_5 &= (b_2)(b_1 b_2 b_2 b_3) = (b_2)(b_1 b_2)(b_2 b_3) = (b_2 b_1)(b_2 b_2 b_3) = \\ &= (b_2 b_1 b_2)(b_2 b_3) = (b_2 b_1 b_2 b_2)(b_3). \end{aligned}$$

Очевидно, $\mathfrak{B}_0 = \{\Lambda, b_2, b_1 b_3\}$ и с ним связаны разложения

$$\begin{aligned} B_2 &= (b_1 b_3)(b_2), \\ B_4 &= (b_1 b_2)(b_1 b_3) = B_1 (b_1 b_3), \\ B_5 &= (b_2)(b_1 b_2)(b_2 b_3) = (b_2) B_1 B_3. \end{aligned}$$

Это дает возможность построить граф $\Gamma(\Sigma)$ (см. рис. 9). $\Gamma(\Sigma)$ содержит ориентированный цикл, порождающий слово B,

$$B = B_1 b_1 b_3 b_2 B_1 B_3,$$

имеющее две расшифровки:

$$B = (B_1 b_1 b_3)(b_2 B_1 B_3), \text{ т. е. } A' = a_4 a_5,$$
$$B = B_1 (b_1 b_3 b_2) B_1 B_3, \text{ т. е. } A'' = a_1 a_2 a_1 a_3.$$

Пример 8. Рассмотрим алфавитное кодирование со схемой

$$\begin{aligned} a_1 &- b_1, \\ a_2 &- b_2 b_1, \\ a_3 &- b_1 b_2 b_2, \\ a_4 &- b_2 b_1 b_2 b_2, \\ a_5 &- b_2 b_2 b_2 b_2. \end{aligned} \qquad (\Sigma)$$

Имеем следующие нетривиальные разложения

$$B_2 = (b_2)(b_1) = (b_2) B_1;$$
$$B_3 = (b_1)(b_2 b_2) = B_1 (b_2 b_2); \quad B_3 = (b_1 b_2)(b_2);$$
$$B_4 = (b_2)(b_1)(b_2 b_2) = (b_2) B_1 (b_2 b_2);$$
$$B_4 = (b_2)(b_1 b_2 b_2) = (b_2) B_3;$$
$$B_4 = (b_2 b_1)(b_2 b_2) = B_2 (b_2 b_2); \quad B_4 = (b_2 b_1 b_2)(b_2);$$
$$B_5 = (b_2)(b_2 b_2 b_2) = (b_2 b_2)(b_2 b_2) = (b_2 b_2 b_2)(b_2).$$

Очевидно, $\mathfrak{B}_0 = \{\Lambda, b_2, b_2 b_2, b_2 b_2 b_2\}$ и с ним связаны

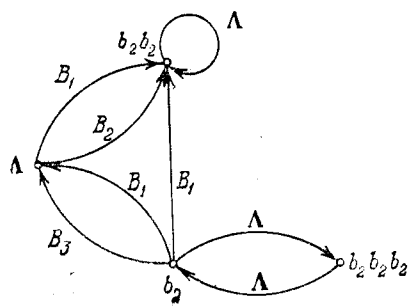


Рис. 10

разложения

$$B_2 = (b_2) B_1;$$
$$B_3 = B_1 (b_2 b_2);$$
$$B_4 = (b_2) B_1 (b_2 b_2); \quad B_4 = (b_2) B_3; \quad B_4 = B_2 (b_2 b_2);$$
$$B_5 = (b_2)(b_2 b_2 b_2) = (b_2 b_2)(b_2 b_2) = (b_2 b_2 b_2)(b_2).$$

Мы получаем граф $\Gamma(\Sigma)$ (см. рис. 10), не содержащий ориентированного цикла, проходящего через вершину Λ.

Следовательно алфавитное кодирование со схемой Σ обладает свойством взаимной однозначности. Данное заключение не может быть выведено из теоремы 2.

§ 3. Об одном свойстве взаимно однозначных кодов *)

В алфавитном кодировании важное место занимают схемы, для которых выполнено свойство взаимной однозначности. Здесь мы докажем несколько результатов для таких кодов.

Пусть задано алфавитное кодирование со схемой Σ

$$\begin{aligned} a_1 &- B_1, \\ &\cdots \\ a_r &- B_r. \end{aligned} \qquad (\Sigma)$$

Пусть, как и раньше, q — число букв в алфавите $\mathfrak{B}$ и $l_i = l(B_i)$ $(i = 1, \ldots, r)$.

Теорема 5 (неравенство Макмиллана [43]). *Если алфавитное кодирование со схемой Σ обладает свойством взаимной однозначности, то*

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant 1. \qquad (2)$$

Доказательство. Рассмотрим всевозможные слова в алфавите $\mathfrak{A}$, имеющие длину n. Все они могут быть порождены выражением

$$(a_1 + \ldots + a_r)^n,$$

если перемножать скобки (например слева) без употребления закона коммутативности и рассматривать произведение

$$a_{i_1} a_{i_2} \ldots a_{i_n}$$

как запись слова в алфавите $\mathfrak{A}$. Здесь, очевидно, символ a_{i_1} будет принадлежать первой скобке, a_{i_2} — второй и т. д.,

*) До сих пор слово «код» употреблялось в общепринятом смысле. Однако в теории кодирования, а еще раньше в технике слово «код» стало трактоваться также и как множество (элементарных) кодов сообщений. Начиная с этого места, слово «код» будет употребляться в обоих смыслах. Из текста, как правило, будет ясно, какой из них имеется в виду. (*Примеч. ред.*)

a_{i_n} — n-й скобке. Мы имеем

$$(a_1 + \ldots + a_r)^n = \sum_{(i_1 i_2 \ldots i_n)} a_{i_1} a_{i_2} \ldots a_{i_n}.$$

Соответствующие этим словам коды получаются, если осуществить замену символов $a_1, \ldots, a_r$ на элементарные коды $B_1, \ldots, B_r$, используя схему алфавитного кодирования. Мы получим

$$(B_1 + \ldots + B_r)^n = \sum_{(i_1 i_2 \ldots i_n)} B_{i_1} B_{i_2} \ldots B_{i_n}. \qquad (3)$$

В силу взаимной однозначности алфавитного кодирования, если $(i_1, \ldots, i_n) \neq (j_1, \ldots, j_n)$, т. е. $a_{i_1} \ldots a_{i_n} \neq a_{j_1} \ldots a_{j_n}$, то

$$B_{i_1} \ldots B_{i_n} \neq B_{j_1} \ldots B_{j_n}.$$

Тождеству (3) соответствует тождество

$$\left(\frac{1}{q^{l_1}} + \ldots + \frac{1}{q^{l_r}}\right)^n = \sum_{(i_1 \ldots i_n)} \frac{1}{q^{l_{i_1} + \ldots + l_{i_n}}}. \qquad (4)$$

Очевидно, что здесь членам с одинаковым знаменателем из правой суммы соответствуют в (3) слова $B_{i_1} B_{i_2} \ldots B_{i_n}$ одинаковой длины. Введем обозначения:

$$t = l_{i_1} + \ldots + l_{i_n},$$

$\nu(n, t)$ — число слов $B_{i_1} B_{i_2} \ldots B_{i_n}$ из (3), имеющих длину t*), и

$$l = \max_{1 \leqslant i \leqslant r} l_i.$$

Мы получим

$$\sum_{(i_1 \ldots i_n)} \frac{1}{q^{l_{i_1} + \ldots + l_{i_n}}} = \sum_{t=1}^{nl} \frac{\nu(n, t)}{q^t}.$$

Из взаимной однозначности алфавитного кодирования вытекает

$$\nu(n, t) \leqslant q^t$$

*) $\nu(n, t) = 0$, если слов длины t в (3) нет.

и, следовательно,

$$\sum_{t=1}^{nl} \frac{\nu(n, t)}{q^t} \leqslant nl.$$

Объединяя последнее неравенство с (4), мы получим

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant \sqrt[n]{nl}.$$

Это неравенство справедливо для любого n. Поэтому оно справедливо и при переходе к пределу при $n \to \infty$.

Окончательно имеем

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant 1.$$

Теорема полностью доказана. Данное утверждение дополняется следующим фактом.

Теорема 6. *Если числа $l_1, \ldots, l_r$ удовлетворяют неравенству* (2) (*неравенству Макмиллана*), *то существует алфавитное кодирование со схемой*

$$\begin{array}{l} a_1 - B'_{1}, \\ \cdots \\ a_r - B'_{r}, \end{array} \qquad (\Sigma')$$

обладающей свойством префикса и удовлетворяющей равенствам

$$l(B'_1) = l_1, \ldots, l(B'_r) = l_r.$$

Доказательство. Можно считать, что $l_1 \leqslant \ldots \leqslant l_r$. Разобьем числа $l_1, \ldots, l_r$ на классы так, что l_i и l_j принадлежат одному классу тогда и только тогда, когда $l_i = l_j$. Пусть μ $(1 \leqslant \mu \leqslant r)$ — число классов, а $\lambda_1, \ldots, \lambda_\mu$, $\nu_1, \ldots, \nu_\mu$ обозначают соответственно представителей и мощности классов. Можно также считать, что

$$\lambda_1 < \lambda_2 < \ldots < \lambda_\mu.$$

Неравенство Макмиллана можно переписать в виде

$$\sum_{i=1}^{\mu} \frac{\nu_i}{q^{\lambda_i}} \leqslant 1.$$

Это неравенство порождает серию вспомогательных неравенств

$$\frac{\nu_1}{q^{\lambda_1}} \leqslant 1 \quad \text{или} \quad \nu_1 \leqslant q^{\lambda_1},$$

$$\frac{\nu_1}{q^{\lambda_1}} + \frac{\nu_2}{q^{\lambda_2}} \leqslant 1 \quad \text{или} \quad \nu_2 \leqslant q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1},$$

. . . . . .

$$\frac{\nu_1}{q^{\lambda_1}} + \frac{\nu_2}{q^{\lambda_2}} + \ldots + \frac{\nu_\mu}{q^{\lambda_\mu}} \leqslant 1 \quad \text{или}$$

$$\nu_\mu \leqslant q^{\lambda_\mu} - \nu_1 q^{\lambda_\mu - \lambda_1} - \nu_2 q^{\lambda_\mu - \lambda_2} - \ldots - \nu_{\mu-1} q^{\lambda_\mu - \lambda_{\mu-1}}.$$

Рассмотрим слова в алфавите $\mathfrak{B}$, имеющие длину λ_1. Так как $\nu_1 \leqslant q^{\lambda_1}$, то можно выбрать из них ν_1 слов длины λ_1. Обозначим их через $B'_1, \ldots, B'_{\nu_1}$. Исключим из дальнейших рассмотрений слова, начинающиеся с $B'_1, \ldots, B'_{\nu_1}$. Далее, возьмем множество слов в алфавите $\mathfrak{B}$, имеющих длину λ_2, которые не начинаются со слов $B'_1, \ldots, B'_{\nu_1}$. Очевидно, что таких слов будет $q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1}$. Так как $\nu_2 \leqslant q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1}$, то из этого множества можно выбрать ν_2 слов — обозначим их через $B'_{\nu_1+1}, \ldots, B'_{\nu_1+\nu_2}$. Исключим из дальнейшего рассмотрения слова, начинающиеся также и с $B'_{\nu_1+1}, \ldots, B'_{\nu_1+\nu_2}$ и т. д. Используя постепенно вспомогательные неравенства, мы построим слова $B'_1, \ldots, B'_r \left(r = \sum_{i=1}^{\mu} \nu_i\right)$. Если их взять в качестве элементарных кодов, то мы получим искомую схему Σ'. Для Σ' выполнено свойство префикса и

$$l(B'_1) = l_1, \ldots, l(B'_r) = l_r.$$

Теорема доказана.

Следствие 1. *Неравенство Макмиллана является необходимым и достаточным условием существования алфавитного кодирования, у которого схема обладает свойством префикса и длины элементарных кодов равны соответственно* $l_1, \ldots, l_r$.

Следствие 2. *Если существует взаимно однозначное алфавитное кодирование с заданными длинами эле-*

ментарных кодов, то существует также алфавитное кодирование со схемой, обладающей свойством префикса, и с теми же длинами элементарных кодов.

Для доказательства следует применить сначала теорему 5, а затем теорему 6.

§ 4. Коды с минимальной избыточностью

Предположим, что задан алфавит $\mathfrak{A} = \{a_1, \ldots, a_r\}$ $(r \geqslant 2)$ и набор вероятностей $p_1, \ldots, p_r \left(\sum_{i=1}^{r} p_i = 1\right)$ появлений символов $a_1, \ldots, a_r$. Пусть, далее, задан алфавит $\mathfrak{B} = \{b_1, \ldots, b_q\}$ $(q \geqslant 2)$. Тогда можно построить целый ряд схем Σ алфавитного кодирования

$$\begin{aligned} a_1 &- B_1, \\ &\cdots \\ a_r &- B_r, \end{aligned} \qquad (\Sigma)$$

обладающих свойством взаимной однозначности. В частности, всегда можно взять в качестве кодов $B_1, \ldots, B_r$ различные слова в алфавите $\mathfrak{B}$ одинаковой длины l, где $l =]\log_q r[$.

Для каждой схемы Σ можно ввести среднюю длину $l_{\text{ср}}$, определяемую как математическое ожидание длины элементарного кода, т. е.

$$l_{\text{ср}} = \sum_{i=1}^{r} p_i l_i, \quad l_i = l(B_i).$$

Очевидно, что величина $l_{\text{ср}}$ $(l_{\text{ср}} \geqslant 1)$ показывает, во сколько раз увеличивается средняя длина слова при кодировании со схемой Σ.

Пример 9. Пусть $r = 4$, $q = 2$ и $p_1 = 0{,}40$, $p_2 = 0{,}25$, $p_3 = 0{,}20$, $p_4 = 0{,}15$.

Рассмотрим две схемы алфавитного кодирования:

$$\begin{aligned} a_1 &- 00, \\ a_2 &- 01, \\ a_3 &- 10, \\ a_4 &- 11, \end{aligned} \qquad (\Sigma') \qquad \begin{aligned} a_1 &- 1, \\ a_2 &- 01, \\ a_3 &- 000, \\ a_4 &- 001. \end{aligned} \qquad (\Sigma'')$$

Они, очевидно, обладают свойством взаимной однозначности. Найдем их средние длины

$$l'_{\text{ср}} = 2, \quad l''_{\text{ср}} = 0{,}40 + 2 \cdot 0{,}25 + 3 \cdot 0{,}35 = 1{,}95.$$

Таким образом, средняя длина может изменяться при переходе от одной схемы алфавитного кодирования к другой. Ввиду этого для данного источника сообщений можно ввести величину l_*, где

$$l_* = \inf_{\Sigma} l_{\text{ср}}^{\Sigma}.$$

(Здесь нижняя грань берется по всем схемам Σ, обеспечивающим свойство взаимной однозначности.)

Легко видеть, что

$$1 \leqslant l_* \leqslant]\log_q r[$$

(верхнюю оценку дает равномерное кодирование). Отсюда видно, что для построения кодов, у которых величина $l_{\text{ср}}$ близка к l_*, можно не учитывать коды с $l_{\text{ср}}$ большим, чем $]\log_q r[$. Значит, для таких схем

$$p_i l_i \leqslant]\log_q r[.$$

Поскольку при вычислении $l_{\text{ср}}$ члены с $p = 0$ не играют роли, то, положив $p_* = \min\limits_{\substack{i \\ p_i \neq 0}} p_i$, имеем

$$l_i \leqslant \frac{]\log_q r[}{p_*}$$

для всех i, для которых $p_i \neq 0$, и, значит, имеется конечное число вариантов значений $l_{\text{ср}}$, для которых $l_* \leqslant l_{\text{ср}} \leqslant]\log_q r[$. Следовательно, величина l_* достигается на некотором Σ и может быть определена как

$$\min_{\Sigma} l_{\text{ср}}^{\Sigma}.$$

Определение. Коды, определяемые схемой Σ с $l_{\text{ср}} = l_*$, называются *кодами с минимальной избыточностью*, или кодами Хафмана [41].

Очевидно, что коды с минимальной избыточностью дают в среднем минимальное увеличение длин слов при соответствующем кодировании. Ввиду этого представляет интерес задача о построении кодов с минимальной избыточностью.

Замечание. В силу следствия 2 из § 3 существует алфавитное кодирование со свойством префикса, дающее коды с минимальной избыточностью. Ввиду этого при построении кодов с минимальной избыточностью

можно ограничиться рассмотрением кодов, обладающих свойством префикса.

Теперь перейдем к вопросу о построении кодов с минимальной избыточностью. Каждому алфавитному кодированию со свойством префикса можно сопоставить кодовое дерево. На примере покажем, каким образом это делается.

Пример 10. Пусть задана схема Σ следующего вида ($r=8$, $q=4$):

$$\begin{array}{ll} a_1 - b_1b_3 & p_1 = 0{,}22 \\ a_2 - b_3 & p_2 = 0{,}20 \\ a_3 - b_1b_1 & p_3 = 0{,}14 \\ a_4 - b_1b_2 & p_4 = 0{,}11 \\ a_5 - b_4b_2b_3 & p_5 = 0{,}10 \\ a_6 - b_1b_4 & p_6 = 0{,}09 \\ a_7 - b_4b_1 & p_7 = 0{,}08 \\ a_8 - b_4b_2b_4 & p_8 = 0{,}06. \end{array}$$

Очевидно, данная схема определяет код со свойством префикса и

$$l_{\text{ср}} = 0{,}20 + 2(0{,}22 + 0{,}14 + 0{,}11 + 0{,}09 + 0{,}08) + \\ + 3\,(0{,}10 + 0{,}06) = 0{,}20 + 2 \cdot 0{,}64 + 3 \cdot 0{,}16 = \\ = 0{,}20 + 1{,}28 + 0{,}48 = 1{,}96.$$

Элементарные коды определяют дерево (см. рис. 11). Концевым вершинам этого дерева соответствуют элементарные коды, определяемые путем (ветвью), идущим из корня, и им приписаны вероятности появления элементарных кодов. Легко видеть, что кодовое дерево, у которого концевым вершинам приписаны вероятности, задает схему алфавитного кодирования со свойством префикса.

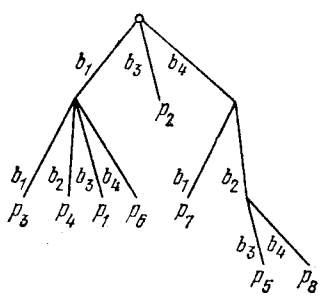


Рис. 11

Сначала рассмотрим частный случай задачи, именно, когда среди чисел $p_1, \ldots, p_r$ имеется не более одного, равного нулю. Без ограничения общности можно считать, что

$$p_1 \geqslant p_2 \geqslant \ldots \geqslant p_r.$$

Лемма 1. *Для кода с минимальной избыточностью из условия* $p_j < p_i$ *следует, что* $l_j \geqslant l_i$.

Доказательство. Предположим противное, т. е. $p_j < p_i$, а $l_j < l_i$. Тогда, если в схеме для кода с минимальной избыточностью:

$$\begin{array}{c} \cdot\ \cdot\ \cdot \\ a_i - B_i, \\ \cdot\ \cdot\ \cdot \\ a_j - B_j, \\ \cdot\ \cdot\ \cdot \end{array} \qquad (\Sigma)$$

поменять местами элементарные коды B_i и B_j, мы получим схему Σ':

$$\begin{array}{c} \cdot\ \cdot\ \cdot \\ a_i - B_j, \\ \cdot\ \cdot\ \cdot \\ a_j - B_i, \\ \cdot\ \cdot\ \cdot \end{array} \qquad (\Sigma')$$

имеющую меньшую среднюю длину $l'_{\text{ср}}$, чем для схемы Σ. В самом деле,

$$l_{\text{ср}} - l'_{\text{ср}} = (p_i l_i + p_j l_j) - (p_i l_j + p_j l_i) =$$
$$= (p_i - p_j)(l_i - l_j) > 0.$$

Последнее противоречит минимальной избыточности Σ. Лемма доказана.

Следствие. *В кодовом дереве для кода с минимальной избыточностью вероятности, приписанные концевым вершинам из* l'*-го яруса, не меньше, чем вероятности, приписанные концевым вершинам из* l''*-го яруса, если* $l'' > l'$.

Далее будем рассматривать конечные деревья с максимальным порядком ветвления (числом исходящих из вершин ребер) q.

Определение. Неконцевая вершина дерева называется *насыщенной*, если ее порядок ветвления равен q. Конечное дерево называется *насыщенным*, если в нем насыщены все неконцевые вершины, за исключением, быть может, одной, лежащей в предпоследнем ярусе, и порядок ветвления этой исключительной вершины равен q_0, где $2 \leqslant q_0 < q$.

В случае, когда в насыщенном дереве нет исключительной вершины, будем считать, что $q_0 = q$ (роль ис-

ключительной вершины в этом случае может играть любая неконцевая вершина из предпоследнего яруса).

Покажем, что по числам r и q число q_0 определяется однозначно. С этой целью заданное насыщенное дерево будет преобразовано в другое насыщенное дерево, имеющее определенную структуру и связанное с исходным деревом только тем, что у него то же число концевых и то же число внутренних вершин. Один шаг преобразования

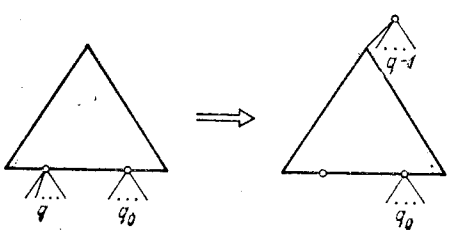


Рис. 12

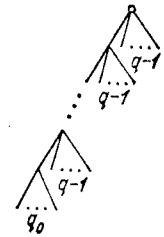


Рис. 13

(см. рис. 12) состоит в перемещении некоторого пучка из q концевых ребер в корень дерева, причем к корню присоединяется одна из концевых вершин (пучок ребер при исключительной вершине не трогается, если даже $q_0 = q$). Такое преобразование выполняется до тех пор, пока это возможно, и в результате получается дерево, изображенное на рис. 13. Из него видно, что число q_0 удовлетворяет уравнению

$$r = t(q-1) + q_0.$$

Вычислим остаток от деления r на $q-1$ и положим

$$q_0 = \begin{cases} q-1, & \text{если остаток равен } 0, \\ q, & \text{если остаток равен } 1, \\ \text{остатку, если он} \geqslant 2. \end{cases} \qquad (5)$$

Лемма 2. *Среди кодов с минимальной избыточностью существует код, кодовое дерево которого является насыщенным.*

Доказательство. Рассмотрим два преобразования кодовых деревьев указанного ниже типа, которые не увеличивают среднюю длину.

1. Удаление ребра в последнем ярусе. Если в некотором пучке последнего яруса кодового дерева содержится ровно одно ребро, то с ним связан эле-

ментарный код $B = B'b$ с вероятностью p, причем слово B' не является префиксом никакого другого элементарного кода. Удаляя данное ребро и перенося вероятность p в вершину, из которой исходило это ребро, получим новое кодовое дерево. Этому преобразованию соответствует переход от схемы Σ к схеме Σ', если заменить в Σ элементарный код B на B'. Очевидно,

$$l'_{\text{ср}} = l_{\text{ср}} - p \leqslant l_{\text{ср}}.$$

2. Перемещение ребер из последнего яруса в вершину кодового дерева, которая не является насыщенной. Пусть кодовое дерево в последнем l-м ярусе содержит не менее двух ребер. Значит, существует ребро, концевой вершине которого приписана вероятность $p\,(p > 0)$, и некоторый элементарный код B. Пусть, далее, имеется вершина, расположенная в l'-м ярусе $(l' \leqslant l-1)$ и не являющаяся насыщенной. Обозначим через B^0 слово, соответствующее этой вершине. Из ненасыщенности вершины следует, что существует символ b_j, для которого $B^0 b_j$ не является префиксом никакого элементарного кода. В этом случае можно упомянутое ребро последнего яруса перенести в данную ненасыщенную вершину по направлению j. Таким образом, заменяя элементарный код B на $B^0 b_j$, мы получаем из схемы Σ схему Σ':

$$l'_{\text{ср}} = l_{\text{ср}} - pl + p\,(l\,(B^0) + 1) \leqslant l_{\text{ср}}.$$

Преобразования 1 и 2 позволяют любой префиксный код из рассматриваемого класса, в том числе и код с минимальной избыточностью, привести к коду, дерево которого является насыщенным, не увеличивая при этом среднюю длину. Лемма доказана.

Пример 11. Используя преобразования 1 и 2, кодовое дерево, изображенное на рис. 11, можно преобразовать в дерево, которое является насыщенным (рис. 14).

Найдем величину $l'_{\text{ср}}$ для этого кода:

$$l'_{\text{ср}} = 0{,}22 + 0{,}20 + 2\,(0{,}14 + 0{,}11 + 0{,}10 + 0{,}09 + \\ + 0{,}08 + 0{,}06) = 0{,}42 + 2 \cdot 0{,}58 = 1{,}58.$$

Средняя длина данного кода меньше исходного.

Замечание. Рассмотрим код с минимальной избыточностью, кодовое дерево которого насыщено. Возьмем пучок ребер последнего яруса, идущих из исключи-

тельной вершины (см. определение на с. 279). Если такой вершины нет, то возьмем произвольный пучок в последнем ярусе. Пусть q_0 — число ребер во взятом пучке, $2 \leqslant q_0 \leqslant q$. В силу следствия вершинам из последнего яруса приписаны вероятности $p_{r-m+1}, \ldots, p_r$, где $m \geqslant q_0$, либо равные им вероятности (это возможно при $p_{r-m} = p_{r-m+1}$). Поэтому путем перестановки элементарных кодов максимальной длины и элементарных кодов, соответствующих одинаковым вероятностям (равным p_{r-m+1}), можно добиться того, чтобы концевым вершинам взятого пучка были приписаны вероятности

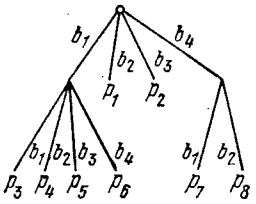


Рис. 14

$$p_{r-q_0+1}, \ldots, p_r.$$

Полученный код будем называть *приведенным*.

Таким образом, имеет место

Теорема 7. *Среди кодов с минимальной избыточностью существует приведенный код, причем q_0 определяется в соответствии с* (5).

Данное утверждение позволяет явно указать набор из q_0 вероятностей $p_{r-q_0+1}, \ldots, p_r$, соответствующий одному из пучков ребер последнего яруса кодового дерева для некоторого кода с минимальной избыточностью. Так, для примера 10 при $r = 8$, $q = 4$ находим

$$8 = 3t + q_0$$

и, пользуясь (5), получаем $t = q_0 = 2$. Значит, выделенному пучку приписаны вероятности p_7 и p_8.

Рассмотрим кодовое дерево произвольного префиксного кода и в нем произвольную концевую вершину. Обозначим приписанную ей вероятность через p. Заменим эту концевую вершину пучком из s ребер ($s \leqslant q$), приписав новым концевым вершинам вероятности $p_{i_1}, \ldots, p_{i_s}$ так, чтобы выполнялось равенство

$$p = p_{i_1} + \ldots + p_{i_s}. \qquad (6)$$

Иными словами, вместо элементарного кода B, соответствующего рассматриваемой вершине, мы включим в код s элементарных кодов $Bb_{j_1}, \ldots, Bb_{j_s}$. Будем говорить в этом случае, что исходный префиксный код преобразуется

в другой префиксный код путем *замены концевой вершины пучком ребер*. Легко видеть, что средняя длина $l_{\text{ср}}$ первого из этих кодов связана со средней длиной $l'_{\text{ср}}$ второго кода равенством

$$l'_{\text{ср}} = l_{\text{ср}} + p. \tag{7}$$

Теорема 8 (редукция). *Пусть один префиксный код преобразуется в другой путем замены в кодовом дереве концевой вершины пучком из s ребер. Далее, пусть концевым вершинам кодового дерева второго кода приписаны вероятности*

$$p_1, \ldots, p_r,$$

причем концевым вершинам указанного пучка — вероятности

$$p_{i_1}, \ldots, p_{i_s} \quad (1 \leqslant i_1 < \ldots < i_s \leqslant r),$$

т. е. концевым вершинам кодового дерева первого кода — вероятности

$$p_1, \ldots, p_{i_1-1}, p_{i_1+1}, \ldots, p_{i_s-1}, p_{i_s+1}, \ldots, p_r, p,$$

где p — определяется равенством (6). *Тогда справедливо следующее.*

I. *Если второй код имеет минимальную избыточность, то первый код также имеет минимальную избыточность.*

II. *Если первый код имеет минимальную избыточность и при этом выполнены условия*

$$p_1 \geqslant \ldots \geqslant p_r,$$

$$s = q_0,$$

где q_0 определяется в соответствии с (5), *и*

$$p_{i_1} = p_{r-q_0+1},$$

$$\ldots\ldots\ldots$$

$$p_{i_s} = p_r$$

$$\left(p = p_{r-q_0+1} + \ldots + p_r\right),$$

то второй код также имеет минимальную избыточность.

Доказательство. Обозначим среднюю длину первого кода через $l'_{\text{ср}}$, а среднюю длину второго

кода — через $l^2_{\text{ср}}$. Для них равенство (7) принимает вид

$$l^2_{\text{ср}} = l^1_{\text{ср}} + p. \tag{8}$$

I. Предположим, что первый код не обладает минимальной избыточностью. Обозначим среднюю длину кода с минимальной избыточностью при тех же вероятностях через $l^3_{\text{ср}}$. По предположению

$$l^3_{\text{ср}} < l^1_{\text{ср}}. \tag{9}$$

Заменив в кодовом дереве этого кода концевую вершину, которой приписана вероятность p, пучком из s ребер, мы получим код со средней длиной $l^4_{\text{ср}}$, удовлетворяющей соотношениям (см. (7), (9) и (8)):

$$l^4_{\text{ср}} = l^3_{\text{ср}} + p < l^1_{\text{ср}} + p = l^2_{\text{ср}}.$$

Но это противоречит минимальной избыточности второго кода. Утверждение I доказано.

II. Предположим, что второй код не обладает минимальной избыточностью. Обозначим среднюю длину кода с минимальной избыточностью при тех же вероятностях через $l^5_{\text{ср}}$. По предположению

$$l^5_{\text{ср}} < l^2_{\text{ср}}. \tag{10}$$

Можно считать, что этот код является префиксным (следствие 2 из § 3) и даже приведенным (теорема 7). Но тогда в его кодовом дереве есть пучок из q_0 ребер, концевым вершинам которого приписаны вероятности $p_{r-q_0+1}, \ldots, p_r$. Заменив этот пучок концевой вершиной, мы получим код со средней длиной $l^6_{\text{ср}}$, удовлетворяющей соотношениям (см. (7), (10) и (8)):

$$l^6_{\text{ср}} = l^5_{\text{ср}} - p < l^2_{\text{ср}} - p = l^1_{\text{ср}}.$$

Но это противоречит минимальной избыточности первого кода. Утверждение II доказано, а с ним доказана и теорема.

Утверждение I показывает, что при построении кода с минимальной избыточностью из более простого кода необходимо, чтобы этот код также имел минимальную избыточность. Однако этого, вообще говоря, недостаточно. Достаточные условия содержит утверждение II.

Теорема 8 дает алгоритм построения кодов с минимальной избыточностью. В этом алгоритме сначала производится мысленное свертывание искомого приведенного кода, в процессе которого один за другим получаются все более простые коды, обладающие минимальной избыточностью, и в конце концов — код из однобуквенных элементарных кодов. Реально это означает преобразование списка вероятностей в соответствии с утверждением II до тех пор, пока не получится список вероятностей, для которого код с минимальной избыточностью находится тривиально. После этого найденный код развертывается в соответствии с утверждением II в последовательность кодов с минимальной избыточностью, которая заканчивается искомым кодом.

При более формальном описании алгоритм удобно разбить на 4 этапа. (В тривиальном случае $r \leqslant q$ остается только 3-й этап.)

1-й этап. С помощью (5) определяется q_0.

2-й этап. Первый шаг. Список вероятностей $p_1, \ldots, p_r$ $(p_1 \geqslant \ldots \geqslant p_r)$ заменяется списком вероятностей $p_1, \ldots, p_{r-q_0}, p$, где $p = p_{r-q_0+1} + \ldots + p_r$, который после упорядочивания превращается в список $p_1, \ldots, p_j, p, p_{j+1}, \ldots, p_{r-q_0}$ $(p_1 \geqslant \ldots \geqslant p_j \geqslant p \geqslant p_{j+1} \geqslant \ldots \geqslant p_{r-q_0})$.

Второй и все последующие шаги 2-го этапа выполняются совершенно аналогично с той лишь особенностью, что для них всегда $q_0 = q$ (после первого шага в кодовом дереве ненасыщенных вершин не остается).

Выполнение 2-го этапа заканчивается, когда число вероятностей в списке становится не больше q.

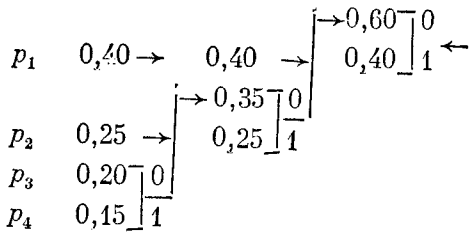
3-й этап. Для списка, содержащего не более, чем q вероятностей, строится префиксный код с минимальной избыточностью. Очевидно, что таким кодом является любой код, состоящий из однобуквенных элементарных кодов.

4-й этап. Первый шаг. В соответствии с теоремой о редукции производится переход от кода с минимальной избыточностью при вероятностях из последнего списка к коду с минимальной избыточностью при вероятностях из предпоследнего списка.

Остальные шаги 4-го этапа выполняются совершенно аналогично и завершаются построением искомого кода с минимальной избыточностью.

Приведем несколько примеров.

Пример 12. Пусть $r=4$, $q=2$ и $p_1=0{,}40$, $p_2=0{,}25$, $p_3=0{,}20$, $p_4=0{,}15$. Возьмем $\mathfrak{B}=\{0, 1\}$. В случае двухбуквенного алфавита $q_0=2$ и процесс построения можно представить следующим образом:



Мы имеем два шага, связанные с редукцией. Квадратными скобками отмечены объединяемые члены. Для построения кодов нужно для каждой скобки выбрать взаимно однозначное соответствие между вероятностями и подмножеством символов из $\mathfrak{B}$.

В нашем случае верхнему числу сопоставим 0, нижнему — 1. Затем осуществляем движение в обратном направлении к символам $p_1, \ldots, p_r$ и, проходя скобки, выписываем соответствующий код. Например, путь

$$0{,}60 - 0{,}35 - 0{,}15 - p_4$$

дает код 001. Таким образом, мы получаем следующую схему:

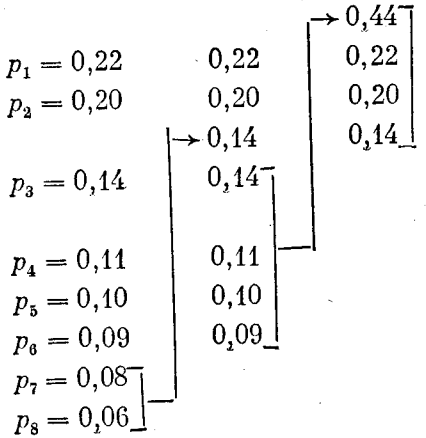
$$\begin{aligned} a_1 &- 1 \\ a_2 &- 01, \\ a_3 &- 000, \\ a_4 &- 001, \end{aligned}$$

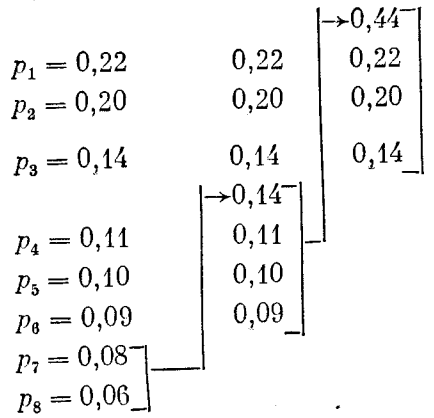
что совпадает со схемой Σ'' на с. 276. Значит, код определяемый Σ'', имеет минимальную избыточность.

Осуществляя редукцию, на каждом шаге производится упорядочение вероятностей по их величине. Это упорядочение оказывается не всегда однозначным, так как возможно появление вероятностей, имеющих одинаковую величину.

Пример 13. Пусть $r=8$, $q=4$ и $p_1=0{,}22$, $p_2=0{,}20$, $p_3=0{,}14$, $p_4=0{,}11$, $p_5=0{,}10$, $p_6=0{,}09$, $p_7=0{,}08$, $p_8=0{,}06$. Возьмем $\mathfrak{B}=\{0, 1, 2, 3\}$. Редукцию можно осуществить

здесь двумя способами:





Отсюда получаем две схемы алфавитного кодирования (нумеруя члены в скобках сверху вниз числами 0, 1, 2, 3)

$a_1 - 1$		$a_1 - 1$
$a_2 - 2$		$a_2 - 2$
$a_3 - 00$		$a_3 - 3$
$a_4 - 01$	(Σ')	$a_4 - 01$ (Σ'')
$a_5 - 02$		$a_5 - 02$
$a_6 - 03$		$a_6 - 03$
$a_7 - 30$		$a_7 - 000$
$a_8 - 31$		$a_8 - 001$

Кодовое дерево для Σ' совпадает с деревом на рис. 14.

В заключение рассмотрим построение кодов с минимальной избыточностью для случая, когда вероятности $p_1, \ldots, p_r$ — произвольны, $p_1 \geqslant \ldots \geqslant p_r$. Если число нулевых вероятностей больше единицы, т. е. $p_{r_0} > 0$ и

$$p_{r_0+1} = \ldots = p_r = 0, \quad r - r_0 > 1,$$

то сначала находят решение для (r_0+1)-буквенного входного алфавита и вероятностей $p_1, \ldots, p_{r_0}, p_{r_0+1}\,(p_{r_0+1} = 0)$. Пусть $B_1, \ldots, B_{r_0}, B_{r_0+1}$ — элементарные коды для кода с минимальной избыточностью. Затем отбрасывают элементарный код B_{r_0+1} и для букв $a_{r_0+1}, \ldots, a_r$ берут в качестве элементарных кодов слова вида

$$B'_{r_0+1} = B_{r_0+1}B^{(r_0+1)},$$
$$\ldots\ldots\ldots\ldots$$
$$B'_r = B_{r_0+1}B^{(r)},$$

где $l\left(B^{(r_0+1)}\right) = \ldots = l\left(B^{(r)}\right)$ и все слова $B^{(r_0+1)}, \ldots, B^{(r)}$ различны. Очевидно, что построенный код имеет минимальную избыточность и удовлетворяет свойству префикса.

§ 5. Самокорректирующиеся коды

Здесь мы рассматриваем один частный случай равномерного кодирования.

Пусть $\mathfrak{A} = \{0, 1\}$ — алфавит, содержащий два символа. Пусть далее $\{A_1, A_2, \ldots, A_s\}$ — множество всех слов $A = \alpha_1 \ldots \alpha_m$ в алфавите $\mathfrak{A}$, имеющих фиксированную длину m. (Здесь $s = 2^m$.)

Предположим, что в канале связи действует источник помех, который в словах из $\{A_1, \ldots, A_s\}$, имеющих длину примерно m, может вызывать ошибки не более чем в p символах. Это значит, что двоичная последовательность, полученная на выходе канала, отличается от двоичной последовательности, поступившей на вход этого канала, не более чем в p позициях. Совершенно ясно, что, если передавать исходное сообщение $\alpha_1 \ldots \alpha_m$ (без предварительного кодирования), то на выходе канала невозможно будет установить, какое сообщение фактически было передано. В связи с этим возникает вопрос, нельзя ли осуществить кодирование слов A из множества $\{A_1, \ldots, A_s\}$, т. е. слов вида $\alpha_1 \ldots \alpha_m$, словами $\beta_1 \ldots \beta_l$ длины l так, чтобы

по коду $\beta_1' \dots \beta_l'$, полученному на выходе канала при передаче кода $\beta_1 \dots \beta_l$, можно было однозначно восстановить этот код, и, значит, исходное сообщение $\alpha_1 \dots \alpha_m$? Коды, обладающие данным свойством, будем называть *самокорректирующимися кодами* относительно рассматриваемого источника помех.

Легко видеть, что существует тривиальное решение задачи. Мы проследим это на простейшем источнике помех, для которого $p = 1$, т. е. для которого возможно только искажение $0 \to 1$ или $1 \to 0$. Искомый самокорректирующийся код получается путем утроения символов исходного кода

$$\alpha_1\alpha_2 \dots \alpha_m \to \alpha_1\alpha_1\alpha_1\alpha_2\alpha_2\alpha_2 \dots \alpha_m\alpha_m\alpha_m.$$

В самом деле, если при передаче этого кода произошла ошибка, то в некоторой группе $\alpha_i\alpha_i\alpha_i$ искажен ровно один символ, а остальные группы переданы без ошибок. Это позволяет методом «голосования» осуществить коррекцию ошибки и восстановить код $(\alpha_1\alpha_1\alpha_1 \dots \alpha_m\alpha_m\alpha_m)$, а значит и исходное сообщение $(\alpha_1 \dots \alpha_m)$.

Тривиальное решение не является корректным, так как длина кода здесь равна $l = 3m$ и мы приходим к кодам, для которых данный источник помех может вызывать бóльшее число ошибок, чем p, и тогда однозначно восстановить исходное сообщение не всегда будет возможно.

Корректное построение самокорректирующихся кодов было осуществлено Хэммингом [40]. Им подробно был разобран случай $p = 1$, к изложению которого мы и перейдем.

Сообщения $\alpha_1 \dots \alpha_m$ кодируются наборами $\beta_1 \dots \beta_l$, где l — длина кода и $l = m + k$. Очевидно, что при наличии данного источника помех возможны следующие варианты получения кодов на выходе (см. рис. 15).

Следовательно, число вариантов равно $l + 1$. Для того чтобы дополнительных разрядов в коде $\beta_1 \dots \beta_l$ хватало для кодировки перечисленных $l + 1$ случаев передачи кода, необходимо, чтобы

$$2^k \geqslant l + 1 \quad \text{или} \quad 2^m \leqslant 2^l/(l + 1).$$

Из этих соображений выберем l как наименьшее целое число, удовлетворяющее неравенству

$$2^m \leqslant 2^l/(l + 1).$$

Дальнейшие построения будут состоять из трех этапов.

I. Построение кодов Хэмминга (описание алгоритма кодирования). Разобьем отрезок натуральных чисел $(1, 2, \ldots, l)$ на k последовательностей следующим образом: пусть V — произвольное натуральное число $(1 \leqslant V \leqslant l)$ и $V_k \ldots V_1$ — его двоичная запись.

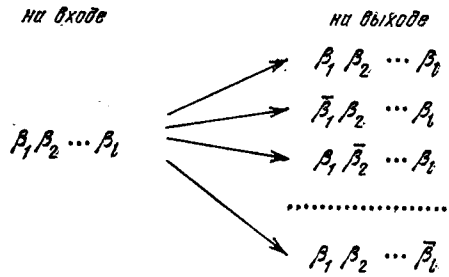


Рис. 15

Последовательность 1, 3, 5, 7, 9, ... содержит все числа V с $V_1 = 1$.

Последовательность 2, 3, 6, 7, 10, ... содержит все числа V с $V_2 = 1$.

Последовательность 4, 5, 6, 7, 12, ... содержит все числа V с $V_3 = 1$.

. . . . . .

Последовательность 2^{k-1}, $2^{k-1}+1$, ... содержит все числа V с $V_k = 1$.

Первыми членами этих последовательностей являются числа

$$1 = 2^0, \quad 2 = 2^1, \ \ldots, \ 2^{k-1},$$

т. е. степени двойки, причем $2^{k-1} \leqslant l$, а $2^k \geqslant l+1$.

Члены β_i набора $\beta_1, \ldots, \beta_l$, у которых индекс i принадлежит множеству $(1, 2, \ldots, 2^{k-1})$, называются *контрольными членами*, остальные — *информационными*. Легко видеть, что контрольных членов будет k, а информационных $l - k = m$.

Сформулируем теперь правило построения набора $\beta_1 \ldots \beta_l$ по набору $\alpha_1 \ldots \alpha_m$. Сначала определяются инфор-

мационные члены

$$\beta_3 = \alpha_1,$$
$$\beta_5 = \alpha_2,$$
$$\beta_6 = \alpha_3,$$
$$\ldots\ldots$$

Таким образом, набор из информационных членов, расположенных в естественном порядке, совпадает с набором $\alpha_1 \ldots \alpha_m$. Далее определяются контрольные члены

$$\beta_1 = \beta_3 + \beta_5 + \beta_7 + \ldots (\mathrm{mod}\ 2),$$
$$\beta_2 = \beta_3 + \beta_6 + \beta_7 + \ldots (\mathrm{mod}\ 2),$$
$$\beta_4 = \beta_5 + \beta_6 + \beta_7 + \ldots (\mathrm{mod}\ 2),$$
$$\ldots\ldots\ldots\ldots$$

Здесь суммирование ведется по последовательностям, построенным выше. В этих формулах правые части, очевидно, состоят из информационных членов, которые нами уже определены. Обозначим через H_l^1 множество всех построенных наборов $\beta_1 \ldots \beta_l$.

II. Обнаружение ошибки в кодах Хэмминга. Пусть $(\beta_1 \ldots \beta_l) \in H_l^1$ и при передаче кода $\beta_1 \ldots \beta_l$ произошла ошибка в S-м члене. Тогда на выходе канала было принято слово $\beta_1' \ldots \beta_l'$, где

$$\beta_1' \ldots \beta_l' = \beta_1 \ldots \bar{\beta}_S \ldots \beta_l.$$

Пусть $S = S_k \ldots S_1$ — запись числа S в двоичном счислении. Покажем, как можно по коду $\beta_1' \ldots \beta_l'$ найти число S.

Рассмотрим число $S' = S_k' \ldots S_1'$, где:

$$S_1' = \beta_1' + \beta_3' + \beta_5' + \beta_7' + \ldots \text{ (1-я последовательность)},$$
$$S_2' = \beta_2' + \beta_3' + \beta_6' + \beta_7' + \ldots \text{ (2-я последовательность)},$$
$$S_3' = \beta_4' + \beta_5' + \beta_6' + \beta_7' + \ldots \text{ (3-я последовательность)},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

Утверждается, что $S = S'$. В самом деле, если $S_1 = 0$, то S не принадлежит 1-й последовательности и тогда

$$\beta_1' + \beta_3' + \beta_5' + \beta_7' + \ldots = \beta_1 + \beta_3 + \beta_5 + \beta_7 + \ldots = 0,$$

поэтому $S_1' = 0$; если $S_1 = 1$, то S принадлежит 1-й последовательности и тогда

$$\beta_1' + \beta_3' + \beta_5' + \beta_7' + \ldots = 1 + \beta_1 + \beta_3 + \beta_5 + \beta_7 + \ldots = 1,$$

поэтому $S_1' = 1$. Таким образом, $S_1 = S_1'$.

Аналогично доказывается, что $S_2 = S'_{2_1}, \ldots, S_k = S'_k$. Отсюда следует, что $S = S'$.

Если при передаче ошибки не произошло, то, очевидно, $S' = 0$. Значит число S' позволяет узнать, произошла ли ошибка при передаче и, если произошла, то найти номер члена S, который исказился помехой. В последнем случае производим коррекцию ошибки: член β'_S заменяем на $\overline{\beta'_S}$.

III. Декодирование. Этот шаг состоит в построении исходного сообщения $\alpha_1 \ldots \alpha_m$ по коду $\beta_1 \ldots \beta_l$. Для этого, очевидно, достаточно взять информационные члены в $\beta_1 \ldots \beta_l$.

Пример 14. Построить самокорректирующийся код для $m = 4$. Наименьшее число l, удовлетворяющее неравенству

$$2^4 \leqslant \frac{2^l}{l+1},$$

будет $l = 7$, и тогда $k = 3$. В соответствии с этапом I получаем самокорректирующийся код. Результат этого построения представлен в табл. 1, в которой контрольные члены помечены звездочкой.

Таблица 1

1*	2*	3	4*	5	6	7	1*	2*	3	4*	5	6	7
0	0	0	0	0	0	0	1	1	1	0	0	0	0
1	1	0	1	0	0	1	0	0	1	1	0	0	1
0	1	0	1	0	1	0	1	0	1	1	0	1	0
1	0	0	0	0	1	1	0	1	1	0	0	1	1
1	0	0	1	1	0	0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1	0	1	0	1	0	1
1	1	0	0	1	1	0	0	0	1	0	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1

В этой таблице сначала в столбцы с номерами 3, 5, 6 и 7 (информационные члены) вписываются сверху вниз наборы 0000, ..., 1111. Затем по формулам

$$\beta_1 = \beta_3 + \beta_5 + \beta_7 \pmod 2,$$
$$\beta_2 = \beta_3 + \beta_6 + \beta_7 \pmod 2,$$
$$\beta_4 = \beta_5 + \beta_6 + \beta_7 \pmod 2$$

заполняются столбцы с номерами 1, 2 и 4.

Пусть на вход канала поступил код 0110011, и в нем источник помех исказил 5-й член ($S=5$). Тогда на выходе мы получим 0110111. Вычислим номер члена, в котором произошла ошибка. Мы имеем

$$S_1' = \beta_1' + \beta_3' + \beta_5' + \beta_7' = 0 + 1 + 1 + 1 = 1,$$
$$S_2' = \beta_2' + \beta_3' + \beta_6' + \beta_7' = 1 + 1 + 1 + 1 = 0,$$
$$S_3' = \beta_4' + \beta_5' + \beta_6' + \beta_7' = 0 + 1 + 1 + 1 = 1.$$

Следовательно, $S'=101$, т. е. $S'=5$. Мы обнаружили член, в котором произошла ошибка, и $S'=S$.

В заключение остановимся на выяснении геометрических свойств кодов Хэмминга.

Будем рассматривать единичный l-мерный куб как метрическое пространство, в котором для любых двух точек $\beta'=(\beta_1', \ldots, \beta_l')$ и $\beta''=(\beta_1'', \ldots, \beta_l'')$ расстояние $\rho(\beta', \beta'')$ определено следующим образом:

$$\rho(\beta', \beta'') = \sum_{i=1}^{l} |\beta_i' - \beta_i''|.$$

Очевидно, $\rho(\beta', \beta'')$ есть число координат, в которых различаются наборы β' и β''.

Теорема 9. *Для любых наборов* β' *и* β'' *таких, что* $\beta' \neq \beta''$ *и* $\beta', \beta'' \in H_l^1$, *имеет место* $\rho(\beta', \beta'') \geqslant 3$.

Доказательство. Утверждение будет доказано, если исключить два случая: а) $\rho(\beta', \beta'')=1$; б) $\rho(\beta', \beta'')=2$. В самом деле, если $\rho(\beta', \beta'')=1$, то возможна единичная ошибка, при которой код β' перейдет в код β'' и, если $\rho(\beta', \beta'')=2$, то существует такой набор β''', что $\rho(\beta', \beta''')=\rho(\beta''', \beta'')=1$, т. е. возможны переходы при единичных ошибках кодов β' и β'' в код β'''.

Следовательно, в обоих случаях на выходе канала возможна ситуация, в которой нельзя установить, какой из кодов β' или β'' фактически был передан, что противоречит самокорректируемости кода H_l^1. Теорема доказана.

Пусть β^0 — некоторая точка единичного l-мерного куба,

Определение. Совокупность $U_l^p(\beta^0)$ точек β таких, что $\rho(\beta^0, \beta) \leqslant p$, называется *шаром* с центром в β^0 и радиусом p.

Определение. Совокупность $V_l^p(\beta^0)$ точек β таких, что $\rho(\beta^0, \beta)=p$, называется *сферой* с центром в β^0 и радиусом p.

Очевидно, что если точка β^0 является «кодовой» точкой, то при передаче по каналу связи, в котором действует источник помех, вызывающий не более p ошибок, точка β^0 перейдет в точку β такую, что

$$\rho(\beta^0, \beta) \leqslant p,$$

т. е. точка β будет принадлежать шару $U_l^p(\beta^0)$ с центром в β^0 и имеющему радиус p. Отсюда вытекает следующий факт.

Теорема 10. *Для того чтобы множество H, принадлежащее единичному l-мерному кубу, было множеством, принадлежащим самокорректирующемуся коду относительно источника помех, вызывающего не более p ошибок, необходимо и достаточно, чтобы для любых β' и β'' ($\beta' \neq \beta''$) из H имело место*

$$\rho(\beta', \beta'') \geqslant 2p + 1.$$

Для доказательства лишь заметим, что условие $\rho(\beta', \beta'') \geqslant 2p + 1$ эквивалентно тому, что шары $U_l^p(\beta')$ и $U_l^p(\beta'')$ с центрами в β' и β'' и имеющие радиусы p, не пересекаются. Последнее означает, что коду, полученному на выходе канала связи, соответствует точка, которая принадлежит ровно одному шару.

Данная теорема дает геометрический подход для построения самокорректирующихся кодов. С ней также связано и следующее утверждение.

Теорема 11. а) *При $l = 2^t - 1$ единичный l-мерный куб может быть разбит в прямую сумму единичных шаров (т. е. шаров с радиусом 1).*

б) *При $l = 2^t$ единичный l-мерный куб может быть разбит в прямую сумму единичных сфер.*

Доказательство. а) В единичном l-мерном кубе с $l = 2^t - 1$ возьмем код Хэмминга H_l^1. Очевидно, имеем

$$k = t, \quad m = 2^t - t - 1.$$

Около каждой точки из H_l^1 опишем шар радиуса 1. Покажем, что система всех таких шаров и задает искомое разбиение. Данные шары попарно не пересекаются (см. теорему 9). Отсюда суммарное число точек, содержащихся в

этой системе шаров, равно

$$(l+1)\,2^m = 2^t 2^{2^t-t-1} = 2^{2^t-1} = 2^l.$$

Значит, эта система шаров содержит все точки единичного l-мерного куба.

б) Пусть $l = 2^t$. При фиксированной последней координате единичный l-мерный куб может быть «разрезан» на два единичных $l-1$-мерных куба. Для этих кубов существует естественное взаимно однозначное соответствие $\beta^0 \rightleftarrows \beta^1$, где

$$\beta^0 = (\beta_1, \ldots, \beta_{l-1}, 0), \quad \beta^1 = (\beta_1, \ldots, \beta_{l-1}, 1).$$

Так как $l-1 = 2^t - 1$, то $l-1$-мерный единичный куб на основании пункта а) может быть разбит в прямую сумму

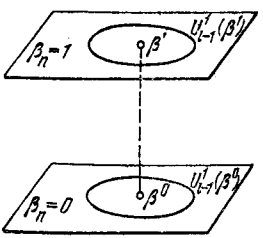


Рис. 16

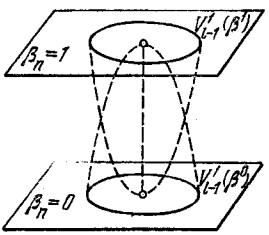


Рис. 17

единичных шаров. Выберем в одном из $l-1$-мерных кубов разбиение в прямую сумму единичных шаров. Разбиение в другом $l-1$-мерном кубе можно построить, используя естественное взаимно однозначное соответствие между этими кубами. Рассмотрим пару соответствующих шаров (см. рис. 16) $U^1_{l-1}(\beta^0)$ и $U^1_{l-1}(\beta^1)$.

Два данных $l-1$-мерных единичных шара можно преобразовать в две l-мерные единичные сферы (рис. 17).

В самом деле, $l-1$-мерные единичные шары являются объединением $l-1$-мерной единичной сферы и своего центра, т. е.

$$U^1_{l-1}(\beta^0) = V^1_{l-1}(\beta^0) \cup \{\beta^0\},$$

$$U^1_{l-1}(\beta^1) = V^1_{l-1}(\beta^1) \cup \{\beta^1\}.$$

Очевидно, что для l-мерных единичных сфер имеют мес-

то равенства

$$V_l^1(\beta^0) = V_{l-1}^1(\beta^0) \cup \{\beta^1\},$$
$$V_l^1(\beta^1) = V_{l-1}^1(\beta^1) \cup \{\beta^0\}.$$

Таким образом, если осуществить это преобразование для всех пар соответствующих шаров разбиений, получим искомое разбиение куба в прямую сумму единичных сфер. Теорема доказана.

Определение. Кодовое множество, принадлежащее l-мерному единичному кубу и являющееся самокорректирующимся для данного источника помех, называется *максимальным*, если оно имеет наибольшую мощность.

Следствие. *При* $l = 2^t - 1$ *код Хэмминга* H_l^1 *является максимальным.*

ЧАСТЬ V

НЕКОТОРЫЕ ПРИЛОЖЕНИЯ К КИБЕРНЕТИКЕ

Глава 1

ДИЗЪЮНКТИВНЫЕ НОРМАЛЬНЫЕ ФОРМЫ

§ 1. Понятие д. н. ф. Проблема минимизации булевых функций

Пусть задан алфавит переменных $\{x_1, \ldots, x_n\}$.

Определение. Выражение

$$K = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r} \quad (i_\nu \neq i_\mu \text{ при } \nu \neq \mu)$$

называется *элементарной конъюнкцией*. Число r называется *рангом элементарной конъюнкции*. По определению считаем константу 1 элементарной конъюнкцией ранга 0.

Определение. Выражение

$$\mathfrak{N} = \bigvee_{i=1}^{s} K_i \quad (K_i \neq K_j \text{ при } i \neq j),$$

где K_i $(i = 1, \ldots, s)$ — элементарная конъюнкция ранга r_i, называется *дизъюнктивной нормальной формой* (*д. н. ф.*).

Очевидно, что д. н. ф. $\mathfrak{N}$ реализует некоторую булеву функцию $f(x_1, \ldots, x_n)$. Из гл. 1 части I следует, что для каждой функции $f(x_1, \ldots, x_n)$ и $f \not\equiv 0$ существует д. н. ф. $\mathfrak{N}$ такая, что

$$f(x_1, \ldots, x_n) = \mathfrak{N}.$$

В качестве такой д. н. ф. можно взять, например, совершенную д. н. ф. для f, а именно:

$$\mathfrak{N} = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n) = 1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Пример 1. Рассмотрим функцию $f(x_1, x_2, x_3)$, заданную таблицей 4. Ее можно представить в виде совершенной д. н. ф.

$$\mathfrak{N}_1 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3.$$

С другой стороны, непосредственной проверкой убеждаемся, что эта функция может быть представлена другой д. н. ф.

$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1.$$

Данный пример показывает, что функция алгебры логики может быть представлена в виде д. н. ф., вообще говоря, не единственным образом. В связи с этим возникает возможность выбора более предпочтительной реализации. Для этого вводится индекс простоты $L(\mathfrak{N})$, характеризующий «сложность» д. н. ф.

Для функционала $L(\mathfrak{N})$ потребуем выполнения следующих аксиом.

I. Аксиома неотрицательности. Для любой д. н. ф. $L(\mathfrak{N}) \geqslant 0$.

II. Аксиома монотонности (относительно умножения). Пусть $\mathfrak{N} = \mathfrak{N}' \vee x_i^{\sigma_i} K'$. Тогда

$$L(\mathfrak{N}) \geqslant L(\mathfrak{N}' \vee K').$$

III. Аксиома выпуклости (относительно сложения). Пусть д. н. ф. $\mathfrak{N}$ разбита в прямую сумму д. н. ф. $\mathfrak{N}_1$ и $\mathfrak{N}_2$, т. е. $\mathfrak{N} = \mathfrak{N}_1 \vee \mathfrak{N}_2$ и $\mathfrak{N}_1$, $\mathfrak{N}_2$ не имеют общих членов. Тогда

$$L(\mathfrak{N}) \geqslant L(\mathfrak{N}_1) + L(\mathfrak{N}_2).$$

IV. Аксиома инвариантности (относительно изоморфизма). Пусть д. н. ф. $\mathfrak{N}'$ получена из д. н. ф. $\mathfrak{N}$ путем переименования переменных (без отождествлений). Тогда

$$L(\mathfrak{N}') = L(\mathfrak{N}).$$

Приведем примеры встречающихся индексов простоты для д. н. ф.

1. $L_{\text{в}}(\mathfrak{N})$ — число букв переменных, встречающихся в записи д. н. ф. $\mathfrak{N}$. Если взять д. н. ф. $\mathfrak{N}_1$ и $\mathfrak{N}_2$ из примера 1, то $L_{\text{в}}(\mathfrak{N}_1) = 15$ и $L_{\text{в}}(\mathfrak{N}_2) = 3$, т. е. в смысле этого индекса д. н. ф. $\mathfrak{N}_2$ проще, чем д. н. ф. $\mathfrak{N}_1$.

2. $L_{\text{к}}(\mathfrak{N})$ — число элементарных конъюнкций, входящих в $\mathfrak{N}$. Для д. н. ф. $\mathfrak{N}_1$ и $\mathfrak{N}_2$, очевидно, $L_{\text{к}}(\mathfrak{N}_1) = 5$ и $L_{\text{к}}(\mathfrak{N}_2) = 2$, т. е. в смысле этого индекса д. н. ф. $\mathfrak{N}_2$ проще, чем д. н. ф. $\mathfrak{N}_1$.

3. $L_0(\mathfrak{N})$ — число символов $^-$, встречающихся в записи д. н. ф. $\mathfrak{N}$. Для д. н. ф. $\mathfrak{N}_1$ и $\mathfrak{N}_2$, очевидно, $L_0(\mathfrak{N}_1)=7$ и $L_0(\mathfrak{N}_2)=2$, т. е. в смысле этого индекса д. н. ф. $\mathfrak{N}_2$ опять проще, чем д. н. ф. $\mathfrak{N}_1$.

Нетрудно проверить, что каждый из указанных индексов удовлетворяет приведенным выше аксиомам.

Замечание. Пусть $\mathfrak{N}=\mathfrak{N}'\vee K$. Тогда в силу аксиом III и I

$$L(\mathfrak{N})\geqslant L(\mathfrak{N}').$$

Очевидно, что над алфавитом переменных $\{x_1, \ldots, x_n\}$ можно построить 3^n различных элементарных конъюнкций («пустой» конъюнкции сопоставлена константа 1). Отсюда следует, что число д. н. ф. над этим же алфавитом из n букв равно 2^{3^n}. Опираясь на этот подсчет, введем следующее определение.

Определение. Д. н. ф. $\mathfrak{N}$, реализующая функцию $f(x_1, \ldots, x_n)$ и имеющая минимальный индекс $L(\mathfrak{N})$, называется *минимальной относительно* L (*м. д. н. ф. относительно* L).

Поскольку дальнейшее изложение связано главным образом с индексом простоты $L_\text{Б}$, то минимальную д. н. ф. относительно этого индекса будем называть *минимальной д. н. ф.* (*м. д. н. ф.*). Д. н. ф., минимальную относительно индекса $L_\text{к}$, будем называть *кратчайшей*.

Вернемся теперь к примеру 1.

1. Д. н. ф. $\mathfrak{N}_2=\bar{x}_2\bar{x}_3\vee x_1$ является минимальной. В самом деле, функция $f(x_1, x_2, x_3)$, реализуемая этой д. н. ф., существенно зависит от переменных x_1, x_2 и x_3 и потому не может быть представлена д. н. ф., содержащей менее трех букв.

2. Д. н. ф. $\mathfrak{N}_2=\bar{x}_2\bar{x}_3\vee x_1$ является кратчайшей, так как функция $f(x_1, x_2, x_3)$, реализуемая этой д. н. ф., отлична от любой элементарной конъюнкции.

3. Д. н. ф. $\mathfrak{N}_2=\bar{x}_2\bar{x}_3\vee x_1$ минимальна относительно L_0, так как функция $f(x_1, x_2, x_3)$, реализуемая этой д. н. ф., по переменным x_2 и x_3 не возрастает и они обе существенны, а поэтому не может быть представлена д. н. ф., содержащей менее двух отрицаний.

Основной вопрос, который нас будет интересовать в этой главе, это как для произвольной функции алгебры логики $f(x_1, \ldots, x_n)$ построить ее минимальную д. н. ф. относительно L. Эта задача называется *проблемой минимизации булевых функций*. Мы покажем сейчас, что дан-

ная задача допускает тривиальное решение. Сначала в каком-либо порядке строят все д. н. ф. (их 2^{3^n}) над переменными $x_1, \ldots, x_n$

$$\mathfrak{N}_1, \mathfrak{N}_2, \ldots, \mathfrak{N}_{2^{3^n}}.$$

Затем отбирают из этого списка те д. н. ф., которые реализуют функцию $f(x_1, \ldots, x_n)$. Наконец, для выбранных д. н. ф. вычисляют величину индекса простоты и путем сопоставлений находят минимальную (относительно L). Сформулированный алгоритм весьма трудоемок с точки зрения его реализации, так как он основан на «переборе» всех д. н. ф., т. е. требует, вообще говоря, не менее 2^{3^n} более мелких операций. Им нельзя воспользоваться практически уже начиная с $n=3$, а для $n=1$ и $n=2$ проблема тривиальна. Таким образом, следует считать, что алгоритмы «полного перебора», т. е. алгоритмы, подобные по трудоемкости тривиальному алгоритму, перебирающему все д. н. ф., являются запрещенным средством в решении проблемы минимизации.

Мы обращаем внимание, что развиваемая далее теория справедлива для любого индекса простоты и потому начальный этап минимизации одинаков для всех индексов простоты. С другой стороны, для удобства можно считать, что в этих построениях речь идет об индексе L_B, т. е. о построении м. д. н. ф. Поскольку минимальная д. н. ф. относительно одного индекса может не быть минимальной д. н. ф. относительно другого индекса (см. [2, 4, раздел 3]), то теория, построенная для одного индекса, вообще говоря, не годится для другого. Однако то обстоятельство, что эти теории имеют и много общего, оправдывает рассмотрение минимизации для конкретного индекса.

§ 2. Упрощение д. н. ф. и тупиковые д. н. ф. (относительно упрощения)

Пусть $\mathfrak{N}$ — произвольная д. н. ф. и

$$\mathfrak{N} = \mathfrak{N}' \vee K, \quad \mathfrak{N} = \mathfrak{N}' \vee x_i^{\sigma_i} K',$$

где K — некоторая элементарная конъюнкция из $\mathfrak{N}$, $\mathfrak{N}'$ — д. н. ф., образованная из остальных конъюнкций, входящих в $\mathfrak{N}$, $x_i^{\sigma_i}$ — некоторый множитель из K, K' — произведение остальных множителей из K. Рассмотрим два типа преобразований д. н. ф.

I. Операция удаления элементарной конъюнкции. Переход от д. н. ф. $\mathfrak{N}$ к д. н. ф. $\mathfrak{N}'$ — преобразование, осуществляемое путем удаления элементарной конъюнкции K. Данное преобразование определено тогда и только тогда, когда $\mathfrak{N}' = \mathfrak{N}$.

II. Операция удаления множителя. Переход от д. н. ф. $\mathfrak{N}$ к д. н. ф. $\mathfrak{N}' \vee K'$ — преобразование, осуществляемое путем удаления множителя $x_i^{\sigma_i}$. Данное преобразование определено тогда и только тогда, когда

$$\mathfrak{N}' \vee K' = \mathfrak{N}.$$

Определение. Д. н. ф. $\mathfrak{N}$, которую нельзя упростить при помощи преобразований I и II (их применить нельзя), называется *тупиковой д. н. ф.* (*т. д. н. ф.*) (относительно преобразований I и II).

Пример 2. Очевидно, д. н. ф. $\mathfrak{N} = x_1 \vee \bar{x}_2\bar{x}_3$ будет тупиковой относительно данных преобразований.

На основе этих двух преобразований можно сформулировать алгоритм упрощения д. н. ф. (Этот алгоритм легко усматривается, и в силу того, что $L(\mathfrak{N}') \leqslant L(\mathfrak{N})$ и $L(\mathfrak{N}' \vee K') \leqslant L(\mathfrak{N})$, он является разновидностью алгоритма наискорейшего спуска. Легко видеть также, что среди тупиковых д. н. ф. функции $f(x_1, \ldots, x_n)$ всегда содержатся и минимальные, может быть, правда, не все.)

1. Выбирается какая-нибудь д. н. ф. для функции $f(x_1, \ldots, x_n)$ в качестве исходной. Таковой можно взять, например, совершенную д. н. ф., так как существует простой способ ее построения.

Таблица 1

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1	1	0	0	1
0	0	1	1	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1

2. Осуществляется упорядочивание в исходной д. н. ф. слагаемых и в каждом слагаемом — множителей. Это упорядочение можно задать записью д. н. ф.

Пример 3. Рассмотрим функцию $f(x_1, x_2, x_3)$ (см. табл. 1). В качестве исходной д. н. ф. для нее возьмем

совершенную д. н. ф. и выберем два упорядочения: одно естественное и второе специальное:

$$\mathfrak{N}' = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3,$$
$$\mathfrak{N}'' = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_3\bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3.$$

3. Затем производится просмотр записи д. н. ф. (слева направо); для очередного члена K_i $(i = 1, \ldots, s)$ сначала пробуют применить операцию удаления элементарной конъюнкции K_i; если это невозможно, то просматривают члены $x_{i_\nu}^{\sigma_\nu}$ конъюнкции K_i слева направо $(\nu = 1, \ldots, r)$

$$K_i = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r}$$

и применяют операцию удаления множителя $x_{i_\nu}^{\sigma_\nu}$ до тех пор, пока это удается.

После этого переходят к следующей элементарной конъюнкции.

Закончив обработку последней элементарной конъюнкции, еще раз просматривают полученную д. н. ф. слева направо и пробуют применять операцию удаления элементарной конъюнкции *).

В результате этого мы получаем искомую д. н. ф. Очевидно (с учетом того, что будет изложено в § 4), имеет место следующий факт.

Теорема 1. *Д. н. ф., полученная в результате применения алгоритма упрощения, является тупиковой д. н. ф. (относительно преобразований* I *и* II).

Пример 4. Для функции $f(x_1, x_2, x_3)$, заданной табл. 1, возьмем в качестве исходной д. н. ф. совершенную д. н. ф. Рассмотрим ее упорядочение, задаваемое записью $\mathfrak{N}'$, и проследим работу алгоритма.

1. Конъюнкция $\bar{x}_1\bar{x}_2\bar{x}_3$, очевидно, не может быть удалена. Однако можно удалить множитель $\bar{x}_1$, так как

$$\bar{x}_2\bar{x}_3 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3.$$

В результате мы получаем конъюнкцию $\bar{x}_2\bar{x}_3$, из которой уже нельзя выбросить ни одного множителя.

2. Конъюнкцию $\bar{x}_1x_2x_3$ удалить также нельзя. Легко видеть, что из нее множитель $\bar{x}_1$ удалить невозможно, в то время как операция удаления множителя $\bar{x}_2$ применима.

*) Необходимость вторичного просмотра можно проиллюстрировать примером (см. табл. 3).

Мы получаем конъюнкцию $\bar{x}_1x_3$, которую упростить путем удаления множителей невозможно.

3. Конъюнкция $\bar{x}_1x_2x_3$ может быть удалена, так как

$$\bar{x}_1x_3 = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3.$$

4. Конъюнкция $x_1\bar{x}_2\bar{x}_3$ также может быть удалена (см. п. 1).

5. Конъюнкцию $x_1x_2\bar{x}_3$ удалить нельзя. Однако, возможно выбросить множитель x_2. В результате мы получим конъюнкцию $x_1\bar{x}_3$, из которой уже нельзя выбросить ни одного множителя.

6. Конъюнкция $x_1x_2x_3$, очевидно, удалена не может быть. Из нее можно удалить множитель x_1. Мы получаем конъюнкцию x_2x_3, которую упростить путем удаления множителей невозможно. Мы получаем д. н. ф. $\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_3 \vee x_2x_3$. Вторичный просмотр этой д. н. ф. с целью удаления конъюнкций упрощений не дает. Следовательно, д. н. ф. $\mathfrak{N}_1$, где

$$\mathfrak{N}_1 = \bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_3 \vee x_2x_3,$$

является результатом применения алгорима упрощения. Приведенные расчеты можно сделать так, как указано в табл. 2.

Для той же функции возьмем другую упорядоченность ее совершенной д. н. ф.:

$$\mathfrak{N}'' = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_3\bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3.$$

В табл. 3 приводятся основные этапы работы алгоритма для этого случая. Следовательно, в этом случае в качестве результата применения алгоритма упрощения получаем д. н. ф.

$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1x_2.$$

Из данного примера вытекает, что результат применения алгоритма упрощения зависит от выбора упорядочения исходной д. н. ф.; так, например,

$$L_B(\mathfrak{N}_1) = 8, \quad L_B(\mathfrak{N}_2) = 6, \quad \text{или} \quad L_B(\mathfrak{N}_1) \neq L_B(\mathfrak{N}_2).$$

Тупиковые д. н. ф. могут иметь различную сложность и, в частности, отличаться от минимальных. В связи с этим возникает вопрос, возможно ли для любой функции $f(x_1, \ldots, x_n)$, исходя из некоторого упорядочивания, получать, применяя алгоритм упрощения, минимальную д. н. ф. Ответ на это дает следующая теорема.

Таблица 2

№ шага	Д.н.ф. и рассматриваемый порядок	Исследуемая конъюнкция	Вид операции
1	$\bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3$	$\bar{x}_1x_2x_3$	удаление $\bar{x}_1$
2	$\bar{x}_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3$	$x_1\bar{x}_2x_3$	удаление $\bar{x}_2$
3	$\bar{x}_2x_3 \vee x_1x_3 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3$	$x_1x_2\bar{x}_3$	удаление $\bar{x}_1\bar{x}_2x_3$
4	$\bar{x}_2x_3 \vee x_1x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3$	$x_1\bar{x}_2\bar{x}_3$	удаление $\bar{x}_1x_2\bar{x}_3$
5	$\bar{x}_2x_3 \vee x_1x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3$	$x_1x_2\bar{x}_3$	удаление x_2
6	$x_2x_3 \vee x_1x_3 \vee x_1\bar{x}_3 \vee x_1x_2x_3$	$x_1x_2x_3$	удаление x_1
7	$x_2x_3 \vee x_1x_3 \vee x_1\bar{x}_3 \vee x_2x_3$		
	Вторичный просмотр ничего не дает	Алгоритм окончен	

Таблица 3

№ шага	Д.н.ф. и рассматриваемый порядок	Исследуемая конъюнкция	Вид операции
	Первый просмотр д. н. ф.		
1	$\bar{x}_1\bar{x}_2x_3 \vee \bar{x}_3x_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_3x_1x_2 \vee \bar{x}_1x_2x_3$	$\bar{x}_1x_2x_3$	удаление $\bar{x}_1$
2	$\bar{x}_2x_3 \vee \bar{x}_3x_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_3x_1x_2 \vee \bar{x}_1x_2x_3$	$x_3\bar{x}_1x_2$	удаление x_3
3	$\bar{x}_2x_3 \vee \bar{x}_1x_2 \vee x_2\bar{x}_1x_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_3x_1x_2 \vee \bar{x}_1x_2x_3$	$x_2\bar{x}_1x_3$	удаление x_2
4	$\bar{x}_2x_3 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee \bar{x}_1x_2x_3$	$x_1x_2x_3$	удаление x_1
5	$\bar{x}_2x_3 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_3x_1x_2 \vee \bar{x}_1x_2x_3$	$\bar{x}_3x_1x_2$	удаление $\bar{x}_3$
6	$\bar{x}_2x_3 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2x_3$	удаление $x_1\bar{x}_2x_3$
	Второй просмотр д. н. ф.		
7	$\bar{x}_2x_3 \vee \bar{x}_1x_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2$	$\bar{x}_2x_3$	неприменимы
8	$x_2\bar{x}_3 \vee \bar{x}_1x_2 \vee x_1\bar{x}_3 \vee x_2x_3 \vee x_1x_2$	$\bar{x}_1x_2$	удаление $\bar{x}_1x_2$
9	$x_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2$	$x_1\bar{x}_3$	неприменимы
10	$x_2\bar{x}_3 \vee \bar{x}_1x_2 \vee x_2x_3 \vee x_1x_2$	x_2x_3	удаление x_2x_3
11	$x_2\bar{x}_3 \vee x_1\bar{x}_3 \vee x_1x_2$	x_1x_2	неприменимы
12	$x_2\bar{x}_3 \vee x_1\bar{x}_3 \vee x_1x_2$	Алгоритм окончен	

Теорема 2. *Пусть* $f(x_1, \ldots, x_n)$ — *произвольная булева функция* ($f \not\equiv 0$) *и* $\mathfrak{N} = \bigvee_{i=1}^{s} K_i$ — *ее произвольная тупиковая д. н. ф.* (*относительно операций* I *и* II); *тогда существует такое упорядочение совершенной д. н. ф., из которого при помощи алгоритма упрощения получается тупиковая д. н. ф.* $\mathfrak{N}$.

Доказательство. Возьмем совершенную д. н. ф. для функции $f(x_1, \ldots, x_n)$ с естественным порядком членов и множителей

$$\mathfrak{N}^0 = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Пусть $x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}$ — ее произвольный член. Так как $f(\sigma_1, \ldots, \sigma_n) = 1$, то существует по крайней мере одна конъюнкция K_i, $i = i\ (\sigma_1, \ldots, \sigma_n)$, из тупиковой д. н. ф. такая, что

$$K_i(\sigma_1, \ldots, \sigma_n) = 1,$$

откуда следует, что $K_i = x_{i_1}^{\sigma_{i_1}} \& \ldots \& x_{i_r}^{\sigma_{i_r}}$. В члене $x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}$ выберем порядок множителей так, чтобы сначала следовали множители, не входящие в K_i, а затем — в произвольном порядке множители из K_i. Следовательно,

$$x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n} = K_\sigma \cdot K_{i(\sigma)} \quad (\sigma = (\sigma_1, \ldots, \sigma_n)).$$

Мы получим некоторое упорядочение совершенной д. н. ф., характеризуемое записью $\mathfrak{N}'$. Легко видеть, что алгоритм упрощения в д. н. ф. $\mathfrak{N}'$ для каждой конъюнкции $K_\sigma \cdot K_{i(\sigma)}$ приведет к одному из исходов: либо ее удалит, либо преобразует в конъюнкцию $K_{i(\sigma)}$. Отсюда д. н. ф. $\mathfrak{N}_1'$ являющаяся результатом работы алгоритма, состоит только из элементарных конъюнкций, входящих в д. н. ф. $\mathfrak{N}$. С другой стороны, в силу тупиковости д. н. ф. $\mathfrak{N}$ должно быть

$$\mathfrak{N}_1' = \mathfrak{N}.$$

Следствие. *В силу того, что среди тупиковых д. н. ф. содержатся обязательно и минимальные относительно* L (*не обязательно все*), *алгоритм упрощения при*

надлежащем выборе упорядочения совершенной д. н. ф. позволяет находить и минимальную д. н. ф.

Замечание. Из доказательства теоремы видно, что для построения всех тупиковых д. н. ф. при помощи алгоритма упрощения из совершенной д. н. ф. достаточно при естественном порядке конъюнкций варьировать только порядок множителей в конъюнкциях.

Таким образом, для построения минимальной д. н. ф. следует перебрать все указанные упорядочения совершенной д. н. ф. $\mathfrak{N}' = \bigvee_{i=1}^{s} K_i'$ и для каждого из них произвести вычисление на основе алгоритма упрощения. Это дает возможность оценить трудоемкость такой процедуры минимизации.

Как видно из определения, однократное применение алгоритма упрощения достаточно просто и содержит $L_к(\mathfrak{N}')$ проверок возможности удаления конъюнкции, $L_Б(K_i')$ проверок возможности удаления множителя из $K_i'(i = 1, \ldots, s')$ и при вторичном просмотре не более $L_к(\mathfrak{N}')$ проверок возможности удаления конъюнкций. Следовательно, число проверок не более

$$\sum_{i=1}^{s} L_Б(K_i') + 2L_к(\mathfrak{N}') \leqslant (n+2)\,2^n.$$

Общее число вариантов упорядочения, как это следует из замечания, равно $(n!)^s$ и

$$(n!)^s \leqslant (n!)^{2^n} \leqslant (n^n)^{2^n} = 2^{2^n n \log n}.$$

Таким образом, число проверок для всех вариантов не более, чем

$$2^{2^n n \log n},$$

что значительно меньше, чем 2^{3^n}, т. е. этот алгоритм лучше, чем алгоритм перебора всех д. н. ф. В то же время трудоемкость алгоритма с использованием процедуры упрощений остается весьма значительной.

Остается обсудить еще одну возможность, влияющую на эффективность алгоритма — случайность выбора упорядочения, играющую роль при решении серии задач на минимизацию булевых функций. Разобьем все вышеуказанные упорядочения на «благоприятные» и «неблагоприятные», смотря по тому, дает или нет алгоритм упро-

щения минимальную д. н. ф. Тогда отношение числа благоприятных упорядочений к числу всех рассматриваемых упорядочений равно вероятности построения минимальной д. н. ф. при случайном выборе порядка. Можно было бы попытаться оценить это отношение. Однако мы не будем этого делать ввиду того, что данная величина связана с конкретным алгоритмом, и ее оценка мало что будет говорить о трудоемкости других алгоритмов. Вместо этого мы возьмем в качестве меры трудоемкости алгоритма родственную характеристику — отношение числа $\mu(f)$ минимальных д. н. ф. функции f к числу $\tau(f)$ ее тупиковых д. н. ф., т. е. величину $\mu(f)/\tau(f)$. Оказывается, что существуют функции с большим числом тупиковых д. н. ф. при относительно малом числе минимальных д. н. ф. Можно показать, что существует последовательность функций $\{f_n\}$, для которой [2]

$$\mu(f_n)/\tau(f_n) \to 0.$$

Последнее заставляет думать, что статистические соображения вряд ли что дают для алгоритма упрощения.

§ 3. Постановка задачи в геометрической форме

Обозначим через E^n множество всех наборов $(\alpha_1, \ldots, \alpha_n)$ из 0 и 1. Его можно рассматривать как множество всех вершин единичного n-мерного куба. Поскольку никаких других, кроме упомянутых, точек мы не рассматриваем, постольку множество E^n будем называть n-мерным кубом, а наборы $(\alpha_1, \ldots, \alpha_n)$ — вершинами куба. На рисунках 1—4 изображены проекции, соответственно, 3-мерного, 4-мерного, 5-мерного и 6-мерного кубов на плоскость (на рис. 4 вершины — не наборы, а соответствующие им натуральные числа (см. стр. 10)).

Определение. Пусть $\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_r}$ — фиксированная система чисел из 0 и 1 такая, что $1 \leqslant i_1 < i_2 < \ldots < i_r \leqslant n$. Множество всех вершин $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ куба E^n таких, что

$$\alpha_{i_1} = \sigma_{i_1}, \alpha_{i_2} = \sigma_{i_2}, \ldots, \alpha_{i_r} = \sigma_{i_r},$$

называется $(n-r)$-*мерной гранью*.

Очевидно, что $(n-r)$-мерная грань является $(n-r)$-мерным подкубом куба E^n.

Пусть $f(x_1, x_2, \ldots, x_n)$ — произвольная функция алгебры логики. Сопоставим ей подмножество N_f вершин куба E^n так, что

$$(\alpha_1, \alpha_2, \ldots, \alpha_n) \in N_f$$

тогда и только тогда, когда

$$f(\alpha_1, \alpha_2, \ldots, \alpha_n) = 1.$$

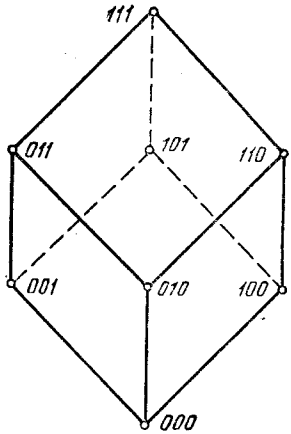


Рис. 1

Ясно, что по подмножеству N_f исходная функция восстанавливается однозначным образом.

Пример 5. Функции $f(x_1, x_2, x_3)$, заданной табл. 4, соответствует множество

$$N_f = \{(0, 0, 0), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\},$$

изображенное на рис. 5.

Возьмем в качестве исходной функции элементарную конъюнкцию $K(x_1, \ldots, x_n)$ ранга r, где

$$K(x_1, \ldots, x_n) = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r}.$$

Легко видеть, что множество N_K, соответствующее конъюнкции K, представляет собой $(n-r)$-мерную грань.

Таблица 4

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1	1	0	0	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	1	1

Число r будем называть также рангом этой грани.

Пример 6. Конъюнкциям

$$K_1(x_1, x_2, x_3) = \bar{x}_2 \& \bar{x}_3,$$

$$K_2(x_1, x_2, x_3) = x_1 \& \bar{x}_2,$$

$$K_3(x_1, x_2, x_3) = x_1$$

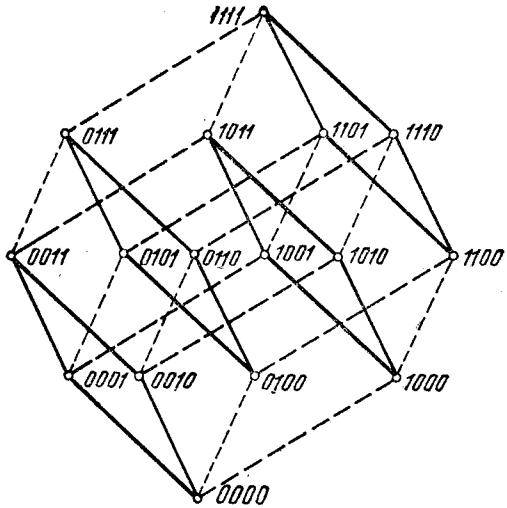

1111
0111
1011
1101
1110
0011
0101
0110
1001
1010
1100
0001
0010
0100
1000
0000

Рис. 2

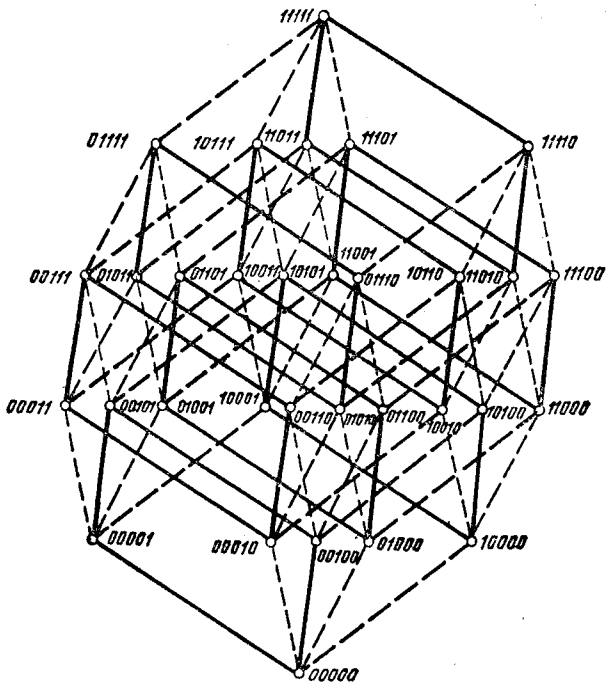

11111
01111
10111
11011
11101
11110
00111
01011
01101
10011
10101
11001
01110
10110
11010
11100
00011
00101
01001
10001
00110
01010
01100
10100
11000
00001
00010
00100
01000
10000
00000

Рис. 3

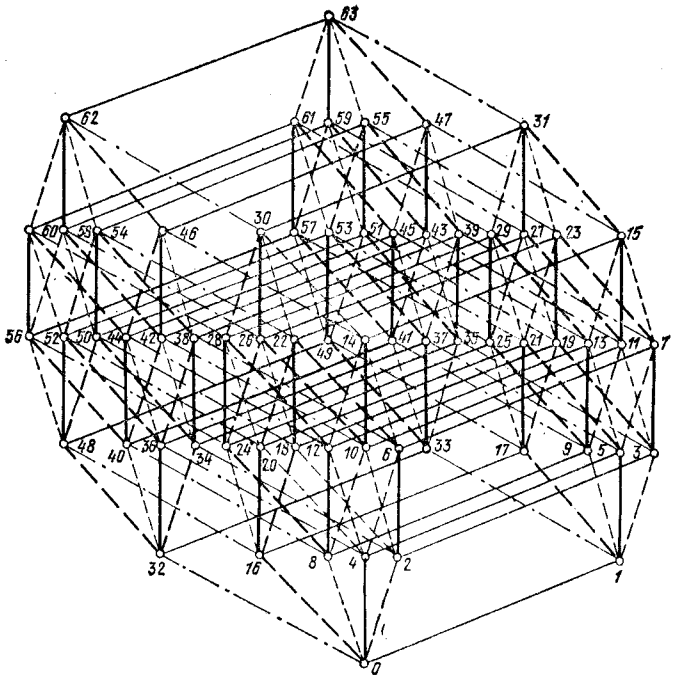


Рис. 4

соответствуют грани

$$N_{K_1} = \{(0, 0, 0), (1, 0, 0)\},$$

$$N_{K_2} = \{(1, 0, 0), (1, 0, 1)\},$$

$$N_{K_3} = \{(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\},$$

имеющие соответственно ранги 2, 2 и 1. Эти грани являются соответственно (см. рис. 5) одномерной гранью (ребром), одномерной гранью (ребром) и двумерной гранью.

Отметим очевидные свойства введенного соответствия $f \rightleftarrows N_f$.

Если

$$f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \vee h(x_1, \ldots, x_n),$$

то:

1) $N_g \subseteq N_f$, $N_h \subseteq N_f$;
2) $N_f = N_g \cup N_h$.

В частности, если функция $f(x_1, \ldots, x_n)$ обладает д. н. ф. $\mathfrak{N}$, где

$$\mathfrak{N} = K_1 \vee \ldots \vee K_s,$$

то из указанных свойств вытекает, что

$$N_{K_i} \subseteq N_f \qquad (i = 1, 2, \ldots, s),$$

т. е. образ конъюнкции K_i, принадлежащей д. н. ф. функции f, является гранью, расположенной внутри множества N_f, и

$$N_f = N_{K_1} \cup N_{K_2} \cup \ldots \cup N_{K_s},$$

т. е. д. н. ф. функции f соответствует покрытие множества N_f гранями $N_{K_1}, \ldots, N_{K_s}$.

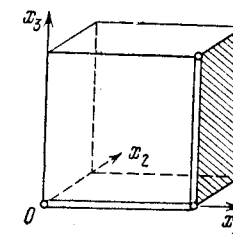


Рис. 5

Нетрудно видеть, что справедливо и обратное утверждение: всякому покрытию множества N_f гранями, расположенными внутри множества N_f, соответствует д. н. ф. $\mathfrak{N}$ функции f.

Пример 7. Мы видели, что

$$\mathfrak{N}_1 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3,$$
$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1$$

являются д. н. ф. функции $f(x_1, x_2, x_3)$ (см. табл. 1). Этим д. н. ф. соответствуют два покрытия множества N_f:

$$N_f = N_{K_1} \cup N_{K_2} \cup N_{K_3} \cup N_{K_4} \cup N_{K_5},$$
$$N_f = N_{K_1^0} \cup N_{K_2^0},$$

где

$$N_{K_1} = \{(0, 0, 0)\}, \quad N_{K_2} = \{(1, 0, 0)\},$$
$$N_{K_3} = \{(1, 0, 1)\}, \quad N_{K_4} = \{(1, 1, 0)\},$$
$$N_{K_5} = \{(1, 1, 1)\}, \; N_{K_1^0} = \{(0, 0, 0), (1, 0, 0)\},$$
$$N_{K_2^0} = \{(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}.$$

Одно из этих покрытий — точечное, второе покрытие состоит из ребра и двумерной грани.

Пусть r_i обозначает ранг грани N_{K_i} (он равен рангу конъюнкции K_i). Число r, где

$$r = \sum_{i=1}^{s} r_{i_1}$$

будем называть *рангом покрытия*. Теперь мы можем дать формулировку геометрической задачи (задачи о покрытии), эквивалентной задаче о минимизации булевой функции: найти для данного множества N_f такое покрытие гранями, принадлежащими N_f,

$$N_f = N_{K_1} \cup N_{K_2} \cup \ldots \cup N_{K_{s_1}},$$

чтобы его ранг r был наименьшим.

Таким образом, можно считать, что задача о минимизации булевой функции имеет две постановки. Одна — в аналитической форме (исходная), вторая — в геометрической форме (задача о покрытии). В связи с этим употребляются два языка: аналитический и геометрический. В ряде случаев используется также комбинированный язык, в котором, например, конъюнкции «называются» гранями, а д. н. ф. — покрытиями.

§ 4. Сокращенная д. н. ф.

Определение. Грань N_K, содержащаяся в N_f, называется *максимальной* (относительно N_f), если не существует грани $N_{K'}$ такой, что:

1. $N_K \subseteq N_{K'} \subseteq N_f$;
2. Размерность грани $N_{K'}$ больше размерности грани N_K.

Пример 8. Пусть $f(x_1, x_2, x_3)$ — функция, заданная табл. 1, и

$$K_1(x_1, x_2, x_3) = \bar{x}_2 \& \bar{x}_3,$$
$$K_2(x_1, x_2, x_3) = x_1 \& \bar{x}_2,$$
$$K_3(x_1, x_2, x_3) = x_1.$$

Тогда грани N_{K_1} и N_{K_3} (см. рис. 5) являются максимальными, а грань N_{K_2} не максимальная для N_f, так как $N_{K_2} \subset N_{K_3}$ и размерность N_{K_3} больше размерности N_{K_2}.

Определение. Конъюнкция K, соответствующая максимальной грани N_K множества N_f, называется *простой импликантой функции* f.

Так как условие $N_K \subseteq N_{K'}$ эквивалентно тому, что все множители из K' содержатся в K, то в определенном смысле из простой импликанты K функции f нельзя удалить ни одного множителя, иначе мы получим (после удаления множителя) конъюнкцию K', для которой $N_{K'} \not\subseteq N_f$.

Отметим очевидное утверждение: каждую грань N_K, $N_K \subseteq N_f$, можно расширить до максимальной грани.

Пусть

$$N_{K_1^0}, N_{K_2^0}, \ldots, N_{K_m^0}$$

— список всех максимальных граней множества N_f. Нетрудно видеть, что

$$N_f = N_{K_1^0} \cup N_{K_2^0} \cup \ldots \cup N_{K_m^0},$$

так как $N_{K_i^0} \subseteq N_f (i = 1, \ldots, m)$ и каждая точка из N_f принадлежит некоторой максимальной грани. Последнее равенство эквивалентно следующему:

$$f = K_1^0 \vee K_2^0 \vee \ldots \vee K_m^0.$$

Определение. Д. н. ф., являющаяся дизъюнкцией всех простых импликант функции f, называется *сокращенной д. н. ф.*

Итак,

$$\mathfrak{N}_C = K_1^0 \vee K_2^0 \vee \ldots \vee K_m^0$$

есть сокращенная д. н. ф. функции f. Как это вытекает из предыдущих рассмотрений, она однозначно определяется по функции f и реализует функцию f.

Пример 9. Для функции f, заданной табл. 4, имеем следующее покрытие, состоящее из всех максимальных граней (см. пример 6)

$$N_f = N_{K_1} \cup N_{K_3}.$$

Ему соответствует сокращенная д. н. ф.

$$\mathfrak{N}_C = \bar{x}_2 \& \bar{x}_3 \vee x_1.$$

Геометрический подход вместе с тем дает и способ построения сокращенной д. н. ф. Однако желательно иметь также и аналитическое решение.

Алгоритм построения сокращенной д. н. ф. Возьмем для функции $f(x_1, \ldots, x_n)$ любую конъюнктивную нормальную форму (например, совершенную к. н. ф.). Затем производим раскрытие скобок, т. е. преобразование типа

$$\& \vee \to \vee \&.$$

После этого в полученном выражении удаляем нулевые члены и ликвидируем поглощаемые и дублирующие члены, т. е. совершаем преобразования вида

$$K_1 K_2 \vee K_1 = K_1,$$

$$K_1 \vee K_1 = K_1.$$

В результате этого мы придем к сокращенной д. н. ф.

Действительно, из любой простой импликанты $x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r}$ функции f в каждую дизъюнкцию исходной конъюнктивной нормальной формы должен входить хотя бы один из членов $x_{i_1}^{\sigma_1}, \ldots, x_{i_r}^{\sigma_r}$ (иначе, положив $x_{i_1} = \sigma_1, \ldots, x_{i_r} = \sigma_r$, мы обратили бы в нуль все члены вида $x_{i_1}^{\overline{\sigma_1}}, \ldots, x_{i_r}^{\overline{\sigma_r}}$, а затем смогли бы подобрать значения остальных переменных так, чтобы некоторая дизъюнкция, не содержащая $x_{i_1}^{\sigma_1}, \ldots, x_{i_r}^{\sigma_r}$, тоже обратилась бы в нуль; но тогда на этом наборе значений переменных $x_1, \ldots, x_n$ функция f обратилась бы в нуль, а простая импликанта $x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r}$ обратилась бы в единицу, что невозможно). Поэтому после раскрытия скобок будут получены все простые импликанты (конъюнкции, не содержащие части сомножителей, получиться не могут, так как соответствующие им грани уже не содержатся в N_f, т. е. каждая такая конъюнкция хотя бы в одном случае равна единице, когда функция f равна нулю). Очевидно, что все остальные конъюнкции, которые будут при этом получены, соответствуют не максимальным граням и, следовательно, поглощаются простыми импликантами.

Пример 10. Возьмем функцию $f(x_1, x_2, x_3)$, заданную табл. 1. Для нее имеем совершенную к. н. ф.

$$f(x_1, x_2, x_3) = (x_1 \vee \overline{x_2} \vee x_3)(\overline{x_1} \vee x_2 \vee \overline{x_3}).$$

Производим раскрытие скобок и упрощения:

$$(x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_3) = x_1\bar{x}_1 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_3 \vee$$
$$\vee x_1x_2 \vee x_2\bar{x}_2 \vee x_2x_3 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3 \vee x_3\bar{x}_3 =$$
$$= \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_1x_2 \vee x_2x_3 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3.$$

Сокращенная д. н. ф. имеет вид

$$\mathfrak{N}_C = \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3.$$

Она легко усматривается и из геометрических соображений и соответствует циклу из ребер (см. рис. 6).

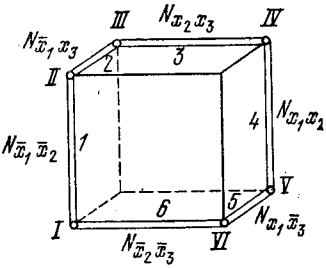


Рис. 6

Пример 11. Рассмотрим функцию $f(x_1, x_2, x_3, x_4, x_5)$:

$$f(\alpha_1, \ldots, \alpha_5) = \begin{cases} 1 \text{ при } 1 \leqslant \alpha_1 + \ldots + \alpha_5 \leqslant 3, \\ 0 \text{ в остальных случаях.} \end{cases}$$

Ее совершенная д. н. ф. есть, очевидно,

$$\mathfrak{N} = \bigvee_{1 \leqslant \sigma_1 + \ldots + \sigma_5 \leqslant 3} x_1^{\sigma_1} \& \ldots \& x_5^{\sigma_5}$$

и имеет сложность $L_к(\mathfrak{N}) = 25$. Каждой грани, содержащейся в N_f, соответствует элементарная конъюнкция, имеющая не менее двух множителей с отрицаниями и не менее одного множителя без отрицаний. В то же время каждой конъюнкции вида $\bar{x}_i\bar{x}_jx_k$, где $i \neq j$, $i \neq k$ и $j \neq k$, соответствует грань, принадлежащая множеству N_f. Отсюда вытекает, что все конъюнкции вида $\bar{x}_i\bar{x}_jx_k$ ($i \neq j$, $i \neq k$, $j \neq k$) являются простыми импликантами функции f и

$$\mathfrak{N}_C = \bigvee_{\substack{(i,j,k) \\ i \neq j, i \neq k, j \neq k}} (\bar{x}_i\bar{x}_jx_k \vee \bar{x}_ix_j\bar{x}_k \vee x_i\bar{x}_j\bar{x}_k)$$

— сокращенная д. н. ф. для f. Очевидно, $L_к(\mathfrak{N}_C) = 3 \cdot \binom{5}{3} = 30$.

Данный пример показывает, что сокращенная д. н. ф. для функции f может иметь большее число членов, чем совершенная д. н. ф.

§ 5. Тупиковость на основе геометрических представлений. Методы построения тупиковых д. н. ф.

Определение. Покрытие множества N_f, состоящее из максимальных (относительно N_f) граней, называется *неприводимым*, если совокупность граней, получающаяся из исходной путем выбрасывания любой грани, не будет покрытием N_f.

Определение. Д. н. ф., соответствующая неприводимому покрытию множества N_f, называется *тупиковой* (в геометрическом смысле).

Пример 12. Для функции $f(x_1, x_2, x_3)$, заданной табл. 1, как видно из рис. 6,

$$N_f = N_{\bar{x}_2\bar{x}_3} \cup N_{\bar{x}_1 x_3} \cup N_{x_1 x_2}$$

является неприводимым покрытием, а

$$\mathfrak{N} = \bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_3 \vee x_1 x_2$$

— тупиковой д. н. ф. (в геометрическом смысле).

Теорема 3. *Понятия тупиковой д. н. ф. относительно преобразований* I *и* II *и тупиковой д. н. ф. в геометрическом смысле эквивалентны.*

В дальнейшем мы будем говорить просто о тупиковых д. н. ф., не указывая определения, из которого мы исходим.

Заметим, что определенные нами д. н. ф.— сокращенная, тупиковая и минимальная находятся в следующем соотношении.

Тупиковая д. н. ф. получается из сокращенной путем отбрасывания (удаления) некоторых членов.

Минимальная (относительно L_B) д. н. ф. является тупиковой.

Среди тупиковых д. н. ф. найдется минимальная (относительно L) д. н. ф.

Рассмотрим теперь более сложный пример на построение тупиковых д. н. ф., используя геометрические соображения.

Пример 13. Пусть $f(x_1, x_2, x_3, x_4)$ задана табл. 5. На рис. 7 изображено множество N_f. В N_f имеются следующие максимальные грани: N_5, N_6, N_7 — ребра, N_1, N_2, N_3, N_4 — грани (двухмерные).

Таким образом, покрытию $N_1 \cup N_2 \cup N_3 \cup N_4 \cup N_5 \cup N_6 \cup N_7$ соответствует сокращенная д. н. ф.

Таблица 5

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	1	1	0	0
1	1	1	1	0
на остальных наборах				1

Две грани N_3 и N_4 входят в любое покрытие, так как только они покрывают соответственно точки (0111) и (1011). Для покрытия точки (0000) нужно взять либо грань N_1, либо грань N_2.

а) Взята грань N_1. Остается покрыть две точки (1100) и (1101), что осуществляется двумя способами: либо взятием ребер N_5 и N_7, либо взятием ребра N_6. Следовательно, получаем два неприводимых покрытия:

$N_1 \cup N_3 \cup N_4 \cup N_5 \cup N_7$,

$N_1 \cup N_3 \cup N_4 \cup N_6$.

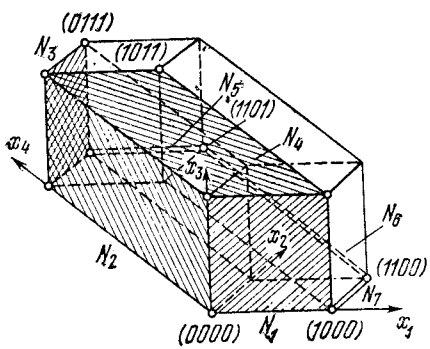


Рис. 7

б) Взята грань N_2. Остается покрыть три точки (1000), (1100) и (1101) — что можно сделать двумя способами: либо взяв ребра N_5 и N_7, либо взяв ребра N_6 и N_7. Здесь мы имеем еще два неприводимых покрытия:

$$N_2 \cup N_3 \cup N_4 \cup N_5 \cup N_7, \quad N_2 \cup N_3 \cup N_4 \cup N_6 \cup N_7.$$

Для построения тупиковых д. н. ф. нужно заметить, что максимальным граням $N_1, \ldots, N_7$ соответствуют простые импликанты

$$K_1 = \bar{x}_2\bar{x}_4, \quad K_2 = \bar{x}_1\bar{x}_2, \quad K_3 = \bar{x}_1x_4, \quad K_4 = \bar{x}_2x_3,$$

$$K_5 = x_2\bar{x}_3x_4, \quad K_6 = x_1x_2\bar{x}_3, \quad K_7 = x_1\bar{x}_3\bar{x}_4.$$

Мы имеем:

$$\mathfrak{N}_1 = \bar{x}_2\bar{x}_4 \vee \bar{x}_1x_4 \vee \bar{x}_2x_3 \vee x_2\bar{x}_3x_4 \vee x_1\bar{x}_3\bar{x}_4,$$
$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_4 \vee \bar{x}_1x_4 \vee \bar{x}_2x_3 \vee x_1x_2\bar{x}_3,$$
$$\mathfrak{N}_3 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_4 \vee \bar{x}_2x_3 \vee x_2\bar{x}_3x_4 \vee x_1\bar{x}_3\bar{x}_4,$$
$$\mathfrak{N}_4 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_4 \vee \bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_3\bar{x}_4;$$
$$L_{\text{Б}}(\mathfrak{N}_2) = 9, \quad L_{\text{Б}}(\mathfrak{N}_1) = L_{\text{Б}}(\mathfrak{N}_3) = L_{\text{Б}}(\mathfrak{N}_4) = 12.$$

Теперь перейдем к построению алгоритма для нахождения всех тупиковых д. н. ф. с использованием геометрических идей.

Алгоритм построения тупиковых д. н. ф. Мы исходим из покрытия множества N_f системой всех его максимальных граней

$$N_{K_1^0}, \ldots, N_{K_m^0}.$$

Пусть $N_f = \{P_1, \ldots, P_\lambda\}$ и P_0 — любая точка *) такая, что $P_0 \notin N_f$ (мы считаем, что $f \not\equiv 1$). Составим таблицу (см. табл. 6), в которой

$$\sigma_{ij} = \begin{cases} 0, & \text{если } P_j \notin N_{K_i^0} \quad (i = 1, \ldots, m), \\ 1, & \text{если } P_j \in N_{K_i^0} \quad (j = 0, 1, \ldots, \lambda). \end{cases}$$

Очевидно, что первый столбец — нулевой, так как $P_0 \notin N_f$, а в каждом столбце, отличном от первого, содержится хотя бы одна единица. Значит, первый столбец отличается от всех остальных.

Для каждого j $(0 \leqslant j \leqslant \lambda)$ найдем множество E_j всех номеров строк, в которых столбец P_j содержит 1 (отличается от столбца P_0).

Пусть $E_j = \{e_{j1}, \ldots, e_{j\mu(j)}\}$. Составим выражение

$$\&_{j=1}^{\lambda} (e_{j1} \vee \ldots \vee e_{j\mu(j)})$$

и произведем преобразование

$$\& \vee \to \vee \&,$$

рассматривая символы e как булевы величины. После этого в полученном выражении ликвидируем поглощаемые или дублирующие члены, т. е. совершаем преобразования

*) Точка P_0 вводится для того, чтобы была видна связь задачи о покрытии с задачей распознавания.

вида

$$A \cdot B \vee A = A,$$
$$A \vee A = A.$$

Мы получим выражение $\vee\,\&'$, являющееся частью выражения $\vee\,\&$. Каждое слагаемое в $\vee\,\&'$ будет определять неприводимое покрытие.

Таблица 6

	P_0	P_1	...	P_j	...	P_λ
$N_{K_1^0}$	σ_{10}	σ_{11}	...	σ_{1j}	...	$\sigma_{1\lambda}$
...	...	...	...	...	...	...
$N_{K_i^0}$	σ_{i0}	σ_{i1}	...	σ_{ij}	...	$\sigma_{i\lambda}$
...	...	...	...	...	...	...
$N_{K_m^0}$	σ_{m0}	σ_{m1}	...	σ_{mj}	...	$\sigma_{m\lambda}$

Действительно, будем рассматривать номера строк как булевы переменные, а каждое подмножество максимальных граней функции f — как набор значений этих переменных: если грань входит в подмножество, то соответствующая ей переменная равна единице, а если нет, то равна нулю. Тогда выражение

$$e_{j1} \vee \ldots \vee e_{j\mu(j)}$$

равно единице тогда и только тогда, когда в подмножество входит максимальная грань, покрывающая точку P_j, а выражение

$$\&_{j=1}^{\lambda} (e_{j1} \vee \ldots \vee e_{j\mu(j)})$$

равно единице тогда и только тогда, когда подмножество максимальных граней покрывает N_f. Поэтому подмножество из максимальных граней, соответствующих переменным из одного слагаемого выражения $\vee\,\&'$, является покрытием N_f. Более того, поскольку выражение $\vee\,\&'$

содержит в качестве слагаемых в точности все свои простые импликанты (ср. со с. 314), а им (как нетрудно заметить) соответствуют неприводимые покрытия N_f, все неприводимые покрытия (и только они) строятся таким способом.

Таблица 7

	0	I	II	III	IV	V	VI
1	0	1	1	0	0	0	0
2	0	0	1	1	0	0	0
3	0	0	0	1	1	0	0
4	0	0	0	0	1	1	0
5	0	0	0	0	0	1	1
6	0	1	0	0	0	0	1

Пример 14. Рассмотрим функцию $f(x_1, x_2, x_3)$ (см. табл. 1). Для нее множество N_f состоит из 6 вершин, которые можно занумеровать числами I, II, ..., VI. Максимальными гранями (см. рис. 6) являются ребра, которые занумерованы арабскими цифрами. Составим таблицу (см. табл. 7). Мы имеем

$$E_{\mathrm{I}} = \{1, 6\}, \quad E_{\mathrm{II}} = \{1, 2\}, \quad E_{\mathrm{III}} = \{2, 3\},$$
$$E_{\mathrm{IV}} = \{3, 4\}, \quad E_{\mathrm{V}} = \{4, 5\}, \quad E_{\mathrm{VI}} = \{5, 6\}.$$

Тогда

$$\bigvee \& = (1 \vee 6)\,(1 \vee 2)\,(2 \vee 3)\,(3 \vee 4)\,(4 \vee 5)\,(5 \vee 6) =$$
$$= (1 \vee 2 \cdot 6)\,(3 \vee 2 \cdot 4)\,(5 \vee 4 \cdot 6) =$$
$$= (1 \cdot 3 \vee 2 \cdot 3 \cdot 6 \vee 1 \cdot 2 \cdot 4 \vee 2 \cdot 4 \cdot 6)\,(5 \vee 4 \cdot 6) =$$
$$= 1 \cdot 3 \cdot 5 \vee 2 \cdot 3 \cdot 5 \cdot 6 \vee 1 \cdot 2 \cdot 4 \cdot 5 \vee \underline{2 \cdot 4 \cdot 5 \cdot 6} \vee$$
$$\vee 1 \cdot 3 \cdot 4 \cdot 6 \vee \underline{2 \cdot 3 \cdot 4 \cdot 6} \vee 1 \cdot \underline{2 \cdot 4 \cdot 6} \vee 2 \cdot 4 \cdot 6 =$$
$$= 1 \cdot 3 \cdot 5 \vee 2 \cdot 3 \cdot 5 \cdot 6 \vee 1 \cdot 2 \cdot 4 \cdot 5 \vee 1 \cdot 3 \cdot 4 \cdot 6 \vee 2 \cdot 4 \cdot 6.$$

Мы получаем пять неприводимых покрытий или пять тупиковых д. н. ф.:

$$\mathfrak{N}_1 = \bar{x}_1\bar{x}_2 \vee x_2x_3 \vee x_1\bar{x}_3,$$
$$\mathfrak{N}_2 = \bar{x}_1x_3 \vee x_2x_3 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3,$$
$$\mathfrak{N}_3 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_1x_2 \vee x_1\bar{x}_3,$$
$$\mathfrak{N}_4 = \bar{x}_1\bar{x}_2 \vee x_2x_3 \vee x_1x_2 \vee \bar{x}_2\bar{x}_3,$$
$$\mathfrak{N}_5 = \bar{x}_1x_3 \vee x_1x_2 \vee \bar{x}_2\bar{x}_3.$$

Две из них $\mathfrak{N}_1$ и $\mathfrak{N}_5$ являются минимальными.

Данный алгоритм оказывается эффективным и для примера, рассмотренного нами на стр. 316. Однако уже даже для функций, зависящих от небольшого числа переменных, алгоритм может оказаться весьма трудоемким и практически непригодным. Это связано с рядом обстоятельств: громоздкостью таблицы, сложностью преобразования $\& \vee \to \vee \&$ и, в конечном счете, большим числом тупиковых д. н. ф. В заключение приведем пример функции от четырех переменных, имеющей много тупиковых д. н. ф.

Пример 15. Рассмотрим функцию $f(x_1, x_2, x_3, x_4)$, где

$$f(x_1, x_2, x_3, x_4) = \overline{x_1x_2x_3x_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4}.$$

Очевидно, данная функция является симметрической и ее конъюнктивная нормальная форма имеет вид

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee x_3 \vee x_4)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4),$$

что позволяет сразу выписать ее сокращенную д. н. ф.

$$\mathfrak{N}_C = x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee x_1\bar{x}_4 \vee \bar{x}_1x_2 \vee x_2\bar{x}_3 \vee x_2\bar{x}_4 \vee$$
$$\vee \bar{x}_1x_3 \vee \bar{x}_2x_3 \vee x_3\bar{x}_4 \vee \bar{x}_1x_4 \vee \bar{x}_2x_4 \vee \bar{x}_3x_4.$$

Отсюда видно, что множество N_f имеет 12 максимальных граней, каждая из которых двумерная. На рис. 8 изображено расположение граней. Мы имеем сферу, образованную максимальными гранями. Занумеруем

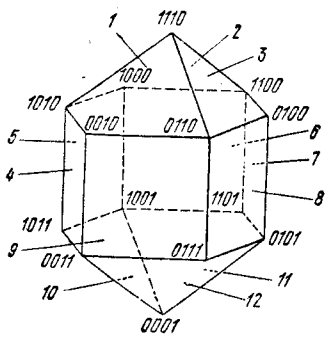


Рис. 8

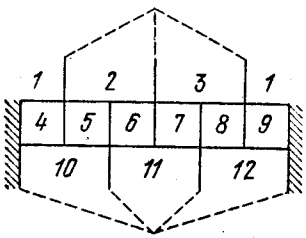


Рис. 9

грани так, как это указано на том же рисунке. Для анализа неприводимых покрытий удобнее использовать развертку сферы на плоскости (см. рис. 9), при этом нужно

иметь в виду, что левый и правый край (помечены штриховкой) должны быть склеены, и вертикальные отрезки сходятся (показано штриховой линией). На развертке грани делятся на верхний слой 1—3, средний пояс 4—9 и нижний слой 10—12. Далее идет перебор неприводимых покрытий в зависимости от фрагмента этого покрытия

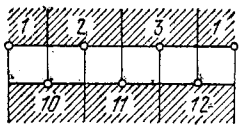


Рис. 10

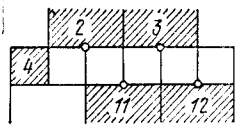


Рис. 11

в среднем поясе. Такой перебор учитывает возможность продолжения этого фрагмента в пределах верхнего и нижнего слоя. Особенно важно иметь в виду запреты граней (помечается минусом) и необходимость покрыть некоторые вершины (помечается жирной точкой). Для каждого случая подсчитывается число вариантов по формуле $a \cdot b \cdot c$, где a — число вариантов в среднем поясе, b — в верхнем слое и c — в нижнем слое.

1) Ни одна грань из среднего пояса не взята. Для покрытия помеченных точек (см. рис. 10) необходимо взять целиком верхний и нижний слои: $a \cdot b \cdot c = 1 \cdot 1 \cdot 1 = 1$.

2) В среднем поясе взята одна грань 4. Для покрытия помеченных точек (см. рис. 11) необходимо взять грани 2, 3 и 11, 12: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

3) В среднем поясе взяты две грани рядом — 4 и 5. Для покрытия помеченных точек (см. рис. 12) необходимо взять грани 3 и 11, 12: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

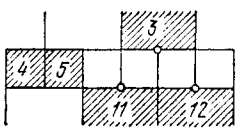


Рис. 12

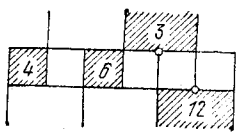


Рис. 13

4) В среднем поясе взяты две грани через одну — 4 и 6. Для покрытия помеченных точек (см. рис. 13) необходимо взять грани 3 и 12: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

5) В среднем поясе взяты две грани через две — 4 и 7. Для покрытия помеченных точек (см. рис. 14) необходимо взять грани 2 и 12. Чтобы получить неприводимое покрытие, из верхнего слоя можно взять только одну грань 1 или 3. Если выбирается грань 1, то из нижнего слоя (грань 10 запрещена) однозначно добавляется грань 11. Наконец, в случае выбора грани 3 из нижнего слоя (грань 11 запрещена) однозначно добавляется грань 10: $a \cdot b \cdot c = 3 \cdot 2 \cdot 1 = 6$.

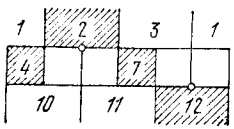


Рис. 14

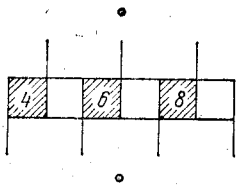


Рис. 15

6) В среднем поясе взяты три грани через одну — 4, 6 и 8. Для покрытия помеченных точек (см. рис. 15) необходимо взять из верхнего слоя любую грань, например, 1. Тогда из нижнего слоя можно взять любую из двух граней 11 и 12 (грань 10 запрещена): $a \cdot b \cdot c = 2 \cdot 3 \cdot 2 = 12$.

7—8) В среднем поясе взяты три грани, из которых две расположены рядом, а третья идет через одну грань (см. рис. 16 и 17). (Оба варианта симметричны, поэтому достаточно разобрать один из них.)

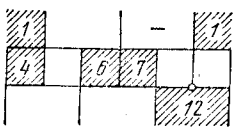


Рис. 16

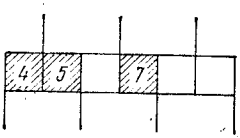


Рис. 17

Для покрытия помеченной точки (см. рис. 16) необходимо взять грань 12. Чтобы получить неприводимое покрытие, можно взять еще только одну грань из верхнего слоя, а именно 1: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

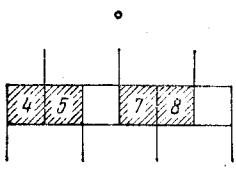


Рис. 18

9) В среднем поясе взяты четыре грани (никакие три из них не идут подряд) — 4,5 и 7,8 (см. рис. 18). Для получения неприводимого покрытия нужно добавить одну произвольную грань из верхнего слоя, например 1. Тогда из нижнего слоя однозначно возьмется еще одна грань 11 (10 и 12 запрещены): $a \cdot b \cdot c = 3 \cdot 3 \cdot 1 = 9$.

Всего мы получаем 58 неприводимых покрытий и, следовательно, 58 тупиковых д. н. ф. и из них 6 — минимальных.

§ 6. Некоторые однозначно получаемые д. н. ф.

Процесс построения минимальных д. н. ф., исходя из совершенной д. н. ф., может быть охарактеризован следующей схемой (см. рис. 19).

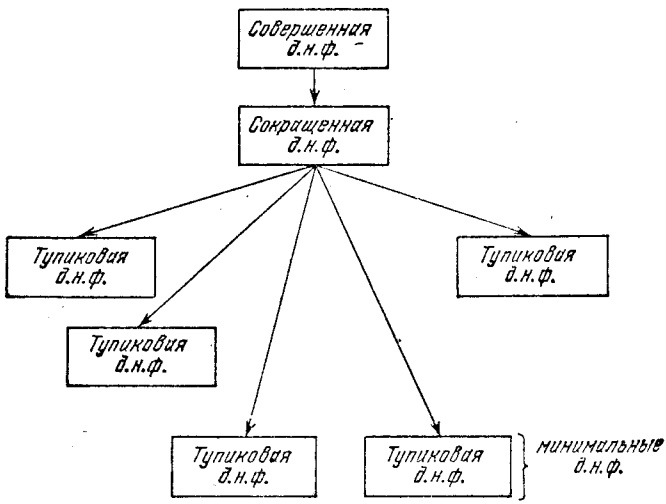


Рис. 19

Сначала получают сокращенную д. н. ф. (при этом на данном шаге возможно усложнение д. н. ф.). Далее однозначный процесс переходит в ветвящийся — процесс построения всех тупиковых д. н. ф. и, наконец, из тупико-

вых д. н. ф. выделяются минимальные д. н. ф. Весьма трудоемким звеном этого процесса является построение тупиковых д. н. ф. (ветвящаяся часть). Его можно пытаться упростить за счет двух обстоятельств.

а) Заранее удалить часть членов сокращенной д. н. ф., не участвующих в построении тупиковых д. н. ф., и тем самым сократить перебор (просматривая подмножества оставшейся части сокращенной д. н. ф.).

б) Произвести удаление части членов сокращенной д. н. ф. так, чтобы оставшаяся часть позволяла построить хоть одну минимальную д. н. ф. Желательно при этом, чтобы данный шаг осуществлялся однозначным образом.

В данном параграфе будут приведены построения двух таких однозначно определяемых д. н. ф.

Определение. Максимальная грань N_K относительно множества N_f называется *ядровой*, если существует такая точка $\tilde{\alpha}$ из N_f, что $\tilde{\alpha} \in N_K$ и $\tilde{\alpha}$ не принадлежит никакой другой максимальной (относительно N_f) грани.

Пример 16. Рассмотрим функцию $f(x_1, x_2, x_3)$ (см. табл. 8). На рис. 20 изображено множество N_f и максимальные грани — ребра N_1, N_2 и N_3. Легко видеть, что N_1 и N_3 являются ядровыми гранями, так как точка $(0, 0, 0)$ покрыта только N_1, а $(1, 1, 1)$ — только N_3.

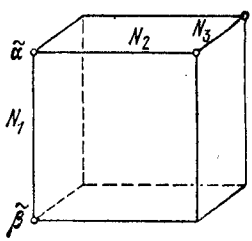


Рис. 20

Таблица 8

x_1	x_2	x_3	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Определение. Множество всех ядровых граней для N_f называется *ядром*.

Очевидно, что в предыдущем примере $\{N_1, N_3\}$ является ядром. Легко видеть, что ядро входит в каждое неприводимое покрытие. Отсюда следует, что грани, покрываемые ядром, не принадлежат ни одному из неприводимых покрытий.

Определение. Д. н. ф. $\mathfrak{N}_{\text{кв}}$, получающаяся из сокращенной путем выбрасывания всех простых импликант, соответствующих максимальным граням, которые покрываются ядром, называется *д. н. ф. Квайна.*

Теорема 4 (Квайн [42]). *Для каждой функции* $f(x_1, \ldots, x_n)$ $(f \not\equiv 0)$, *существует единственная д. н. ф. Квайна.*

Определение д. н. ф. Квайна фактически дает основу для формулировки алгоритма, позволяющего строить д. н. ф. Квайна.

Пример 17. Для функции, заданной табл. 8, имеем сокращенную д. н. ф.

$$\mathfrak{N}_C = \bar{x}_1\bar{x}_2 \vee \bar{x}_2x_3 \vee x_1x_3.$$

Ядро $\{N_1, N_3\}$ (см. с. 325) покрывает грань N_2, которой соответствует простая импликанта $\bar{x}_2x_3$. Таким образом, д. н. ф. Квайна имеет вид

$$\mathfrak{N}_{\text{кв}} = \bar{x}_1\bar{x}_2 \vee x_1x_3.$$

Итак, от сокращенной д. н. ф. путем выбрасывания некоторых импликант возможно перейти к однозначно определенной д. н. ф.— д. н. ф. Квайна, которая реализует ту же функцию и содержит все ее тупиковые д. н. ф.

Следующий шаг в направлении продления однозначной части процесса минимизации связан с определением д. н. ф. типа ΣT.

Определение. Д. н. ф., соответствующая покрытию множества N_f совокупностью всех таких максимальных граней (относительно N_f), которые входят по крайней мере в одно из неприводимых покрытий, называется *д. н. ф. типа* ΣT и обозначается $\mathfrak{N}_{\Sigma T}$.

Очевидно, д. н. ф. $\mathfrak{N}_{\Sigma T}$ получается логическим суммированием (т. е. дизъюнкцией) всех тупиковых д. н. ф. функции f и последующим приведением подобных членов.

Как вытекает из определения, для каждой функции $f(x_1, \ldots, x_n)$ существует единственная д. н. ф. типа ΣT, ее реализующая. Она получается из сокращенной д. н. ф. удалением некоторых членов.

Определение. Пусть $\tilde{\alpha} \in N_f$, тогда совокупность $\Pi_{\tilde{\alpha}}$ всех максимальных граней (относительно N_f), содержащих точку $\tilde{\alpha}$, называется *пучком, проходящим через* $\tilde{\alpha}$.

Определение. Пусть $\tilde{\alpha} \in N_f$ и N_{K0} некоторая максимальная грань такая, что $\tilde{\alpha} \in N_{K0}$. Точка $\tilde{\alpha}$ называется

регулярной точкой (*относительно* N_{K^0} *и* N_f), если существует точка $\tilde{\beta} \in N_f \setminus N_{K^0}$ и $\Pi_{\tilde{\beta}} \subseteq \Pi_{\tilde{\alpha}}$.

Пример 18. Для функции $f(x_1, x_2, x_3)$ (см. табл. 8 и рис. 20) возьмем в качестве точки $\tilde{\alpha}$ вершину $(0, 0, 1)$ и максимальную грань N_2. Очевидно, $\tilde{\alpha} \in N_2$. Покажем, что точка $\tilde{\alpha}$ является регулярной точкой (относительно N_2 и N_f). Пусть $\tilde{\beta} = (0, 0, 0)$. Мы имеем: $\Pi_{\tilde{\alpha}} = \{N_1, N_2\}$, $\Pi_{\tilde{\beta}} = \{N_1\}$ и

$$\Pi_{\tilde{\beta}} \subseteq \Pi_{\tilde{\alpha}}.$$

Определение. Максимальная грань N_{K^0} для N_f называется *регулярной*, если каждая ее точка является регулярной (относительно N_{K^0} и N_f).

В нашем примере, очевидно, N_2 будет регулярной гранью, а N_1 и N_3 не являются регулярными гранями.

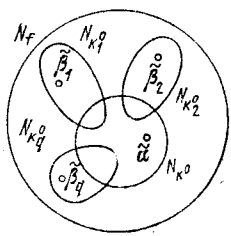


Рис. 21

Теорема 5 (Ю. И. Журавлев [5]). *Для того чтобы простая импликанта* K^0 *функции* f *не принадлежала д. н. ф. типа* ΣT *необходимо и достаточно, чтобы соответствующая максимальная грань* N_{K^0} *была регулярной.*

Доказательство. Необходимость. Пусть K^0 — простая импликанта функции f, K^0 не принадлежит д. н. ф. типа ΣT, а N_{K^0} вопреки утверждению теоремы, не является регулярной гранью. В таком случае существует точка $\tilde{\alpha}$, $\tilde{\alpha} \in N_{K^0}$, которая не регулярна. Обозначим через $\tilde{\beta}_1, \ldots, \tilde{\beta}_q$ точки из множества $N_f \setminus N_{K^0}$ (см. рис. 21):

$$N_f \setminus N_{K^0} = \{\tilde{\beta}_1, \ldots, \tilde{\beta}_q\}.$$

По условию

$$\Pi_{\tilde{\beta}_1} \not\subseteq \Pi_{\tilde{\alpha}}, \ldots, \Pi_{\tilde{\beta}_q} \not\subseteq \Pi_{\tilde{\alpha}},$$

поэтому найдутся грани $N_{K_1^0}, \ldots, N_{K_q^0}$, соответственно принадлежащие пучкам $\Pi_{\tilde{\beta}_1}, \ldots, \Pi_{\tilde{\beta}_q}$ такие, что

$$\tilde{\alpha} \notin N_{K_1^0}, \ldots, \alpha \notin N_{K_q^0}.$$

Очевидно,

$$N_{K^0} \cup N_{K_1^0} \cup \ldots \cup N_{K_q^0} = N_f.$$

Это покрытие позволяет построить неприводимое покрытие множества N_f. При этом могут выброситься некоторые из граней $N_{K_1^0}, \ldots, N_{K_q^0}$, в то же время грань N_{K^0} обязательно войдет в неприводимое покрытие, так как только она одна покрывает точку $\tilde{\alpha}$. Мы получаем, что K^0 входит в некоторую тупиковую д. н. ф. и, следовательно, в д. н. ф. типа ΣT. Полученное противоречие доказывает, что грань N_{K^0} — регулярна.

Достаточность. Пусть N_{K^0} — регулярная грань, покажем, что K^0 не принадлежит д. н. ф. типа ΣT.

Обозначим через $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$ точки из множества N_{K^0}, т. е.

$$N_{K^0} = \{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t\}.$$

В силу регулярности N_{K^0} найдутся точки $\tilde{\beta}_1, \ldots, \tilde{\beta}_t$ из $N_f \setminus N_{K^0}$ такие, что

$$\Pi_{\tilde{\beta}_1} \subseteq \Pi_{\tilde{\alpha}_1}, \ldots, \Pi_{\tilde{\beta}_t} \subseteq \Pi_{\tilde{\alpha}_t}. \qquad (*)$$

Возьмем произвольную тупиковую д. н. ф. $\mathfrak{N}$ функции f и соответствующее ей неприводимое покрытие. Это покрытие

$$N_f = N_1 \cup \ldots \cup N_m,$$

очевидно, покрывает точки $\tilde{\beta}_1, \ldots, \tilde{\beta}_t$ соответственно гранями $N_{i_1}, \ldots, N_{i_t}$ (см. рис. 22). В силу включений $(*)$, эти же грани покрывают точки $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$, т. е.

$$N_{i_1} \cup \ldots \cup N_{i_t} \supseteq N_{K^0}.$$

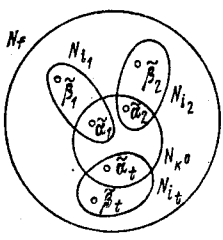


Рис. 22

Поэтому грань N_{K^0} не принадлежит данному неприводимому покрытию, а простая импликанта K^0 — д. н. ф. $\mathfrak{N}$. Теорема полностью доказана.

Данная теорема дает основу для формулировки алгоритма, позволяющего строить д. н. ф. типа ΣT: необходимо из сокращенной д. н. ф. выбросить все конъюнкции, соответствующие регулярным граням,

Пример 19. Для функции $f(x_1, x_2, x_3)$ (см. табл. 8), как видно из предыдущего, имеется одна регулярная грань N_2. Выбросив ее, получим $N_1 \cup N_3$, что дает д. н. ф. $\mathfrak{N}_{\text{ят}}$, совпадающую с единственной тупиковой д. н. ф. этой функции.

Таблица 9

x_1	x_2	x_3	x_4	x_5	f	x_1	x_2	x_3	x_4	x_5	f
0	0	0	0	0	1	0	1	1	1	0	1
0	0	0	0	1	1	1	0	0	0	0	1
0	0	0	1	0	1	1	0	0	1	0	1
0	0	1	0	0	1	1	1	0	0	0	1
0	0	1	1	0	1	1	1	0	1	0	1
0	1	0	0	1	1	на остальных наборах					0
0	1	1	0	0	1						

Возникает вопрос о соотношении д. н. ф. Квайна и д. н. ф. типа ΣТ.

Теорема 6. *Д. н. ф.* $\mathfrak{N}_{\text{ят}}$ *функции* f *получается из д. н. ф. Квайна* $\mathfrak{N}_{\text{кв}}$ *той же функции путем, быть может, выбрасывания некоторых простых импликант.*

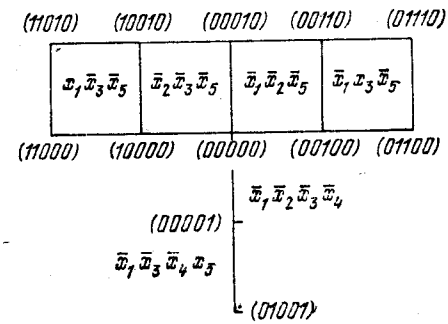


Рис. 23

Доказательство вытекает из очевидного факта: каждая максимальная грань, которая поглощается ядром, является регулярной.

Следующий пример показывает, что д. н. ф. $\mathfrak{N}_{\text{ят}}$ может быть проще, чем д. н. ф. $\mathfrak{N}_{\text{кв}}$.

Пример 20. Возьмем функцию $f(x_1, x_2, x_3, x_4, x_5)$ (см. табл. 9). На рис. 23 представлено расположение мак-

симальных граней для N_f, состоящее из четырех квадратов и двух ребер. Мы имеем для данной функции сокращенную д. н. ф.

$$\mathfrak{N}_C = x_1\bar{x}_3\bar{x}_5 \vee \bar{x}_2\bar{x}_3\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_5 \vee \bar{x}_1x_3\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1\bar{x}_3\bar{x}_4x_5.$$

Ребро $N_{\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4}$ является регулярной гранью, остальные

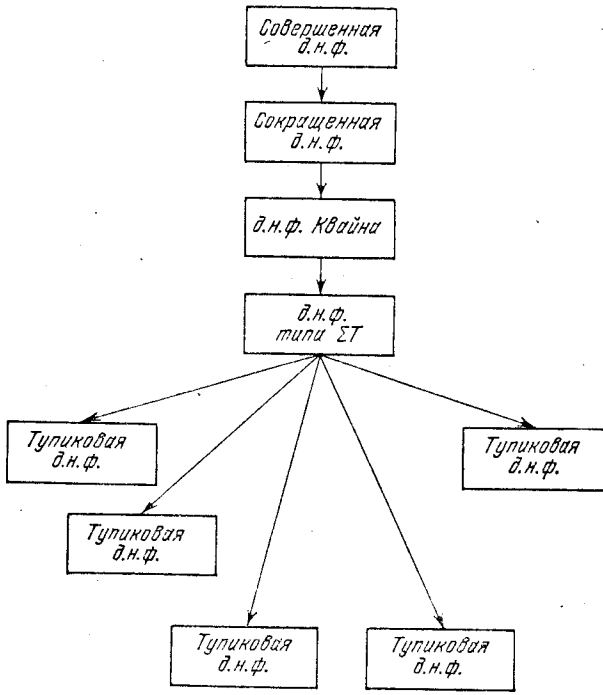


Рис. 24

грани, очевидно, не будут регулярными, поэтому

$$\mathfrak{N}_{\Sigma T} = x_1\bar{x}_3\bar{x}_5 \vee \bar{x}_2\bar{x}_3\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_5 \vee \bar{x}_1x_3\bar{x}_5 \vee \bar{x}_1\bar{x}_3\bar{x}_4x_5.$$

В то же время, так как ядро, состоящее из граней

$$N_{x_1\bar{x}_3\bar{x}_5},\ N_{\bar{x}_1x_3\bar{x}_5},\ N_{\bar{x}_1\bar{x}_3\bar{x}_4x_5},$$

не покрывает ни одной из максимальных граней, то

$$\mathfrak{N}_{\text{кв}} = \mathfrak{N}_C \neq \mathfrak{N}_{\Sigma T}.$$

Таким образом, теперь процесс минимизации можно охарактеризовать схемой (см. рис. 24). Здесь ветвящийся процесс начинается после построения д. н. ф. $\mathfrak{N}_{\text{ЗТ}}$. Дальнейшее продолжение однозначной ветви процесса минимизации, разумеется, возможно. Например, можно дойти до д. н. ф. типа ΣМ (сумма минимальных д. н. ф. с последующим приведением подобных членов). Однако, как мы увидим ниже, это ведет к значительному усложнению алгоритма.

§ 7. Понятие локального алгоритма

Алгоритмы построения д. н. ф. $\mathfrak{N}_{\text{КВ}}$ и д. н. ф. $\mathfrak{N}_{\text{ЗТ}}$ обладают определенной спецификой. Они базируются на специальном изучении покрытия N_f системой всех максимальных граней, в результате которого накапливается определенная информация о каждой максимальной грани. Например, выясняется, является ли грань N_{K^0} ядровой (входит в каждое неприводимое покрытие) или нет; выясняется, является ли грань N_{K^0} регулярной (не входит ни в одно неприводимое покрытие) или нет. Опираясь на эту информацию, далее осуществляется удаление некоторых максимальных граней. Например, в одном случае — покрываемых ядром, в другом случае — регулярных граней.

Для обобщения данных алгоритмов существенную роль играет понятие окрестности порядка u данной максимальной грани N_{K^0} множества N_f.

Определение (индуктивное). Множество $S_0(N_{K^0}) = \{N_{K^0}\}$ называется *окрестностью порядка* 0 максимальной грани N_{K^0}. Пусть определены окрестности порядков $0, 1, \ldots, u-1$ максимальной грани N_{K^0}. Тогда *окрестность* $S_u(N_{K^0})$ *порядка* u максимальной грани N_{K^0} определяется как множество всех максимальных граней из N_f, имеющих непустое пересечение с гранями из $S_{u-1}(N_{K^0})$. Очевидно, что

$$S_0(N_{K^0}) \subseteq S_1(N_{K^0}) \subseteq \ldots \subseteq S_u(N_{K^0}) \subseteq \ldots$$

Пример 21. Возьмем функцию, заданную табл. 1, и в качестве K^0 конъюнкцию $x_1\bar{x}_2$ (см. рис. 6).

Тогда

$$S_0(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}\},$$
$$S_1(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}, N_{\bar{x}_2\bar{x}_3}, N_{\bar{x}_1 x_3}\},$$
$$S_2(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}, N_{\bar{x}_2\bar{x}_3}, N_{\bar{x}_1 x_3}, N_{x_1\bar{x}_3}, N_{x_2 x_3}\},$$
$$S_3(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}, N_{\bar{x}_2\bar{x}_3}, N_{\bar{x}_1 x_3}, N_{x_1\bar{x}_3}, N_{x_2 x_3}, N_{x_1 x_2}\},$$
$$S_u(N_{\bar{x}_1\bar{x}_2}) = S_3(N_{\bar{x}_1\bar{x}_2}), \quad u \geqslant 3.$$

Здесь

$$S_0(N_{\bar{x}_1\bar{x}_2}) \subset S_1(N_{\bar{x}_1\bar{x}_2}) \subset S_2(N_{\bar{x}_1\bar{x}_2}) \subset S_3(N_{\bar{x}_1\bar{x}_2}) = S_4(N_{\bar{x}_1\bar{x}_2}) = \ldots$$

Обозначим через u_0 минимальный порядок окрестности такой, что

$$S_{u_0+1}(N_{K^0}) = S_{u_0}(N_{K^0}).$$

Легко видеть (см. также предыдущий пример), что

$$S_0(N_{K^0}) \subset S_1(N_{K^0}) \subset \ldots \subset S_{u_0}(N_{K^0}) = S_{u_0+1}(N_{K^0}) = \ldots,$$

причем каждая окрестность порядка u $(u > 0)$ содержит по крайней мере одну точку, не входящую в окрестность порядка $u-1$, если $u \leqslant u_0 - 1$. Отсюда $u_0 \leqslant 2^n$.

Зафиксируем теперь параметр u и будем изучать покрытие множества N_f на основе сведений об ее окрестностях порядка u. Это изучение аналогично попыткам составления плана местности на основе сведений об участках местности, которые человек видит из определенных ее точек.

Наши рассмотрения мы начнем с примера, который пояснит смысл локального изучения покрытий. Пусть

$$K_1^0 \vee \ldots \vee K_m^0$$

— сокращенная д. н. ф. функции f, а

$$N_{K_1^0} \cup \ldots \cup N_{K_m^0}$$

— покрытие (максимальными гранями) множества N_f.

Определение. Грани $N_{K_i^0}$ и $N_{K_j^0}$ называются *связными*, если существует такое u, что $N_{K_j^0} \in S_u\left(N_{K_i^0}\right)$.

Легко видеть, что множество всех граней $\{N_{K_i}\}$ однозначным образом разбивается на связные компоненты, т. е. на такие подмножества, в которых каждая пара граней связна, а грани из разных подмножеств не связны.

Возникает вопрос, как для произвольной грани N_{K^0} найти компоненту связности, которой она принадлежит.

Отметим конъюнкцию K^0 и начнем просмотр конъюнкций в порядке их следования в сокращенной д. н. ф. Если $N_{K^0} \in S_1\left(N_{K_1^0}\right)$, то отмечаем конъюнкцию K_1^0, если $N_{K^0} \notin S_1\left(N_{K_1^0}\right)$, то конъюнкцию не отмечаем. Затем переходим к K_2^0. Конъюнкцию K_2^0 отмечаем тогда и только тогда, когда $S_1\left(N_{K_2^0}\right)$ содержит грани для отмеченных конъюнкций, и т. д. В результате этого мы просмотрим всю сокращенную д. н. ф. и отметим некоторые конъюнкции. Затем начинаем просмотр сокращенной д. н. ф. повторно. Если при этом множество отмеченных конъюнкций возрастает, то начинаем следующий просмотр сокращенной д. н. ф., и т. д. Мы придем к случаю, когда очередной просмотр (а их меньше m) не увеличит множества отмеченных конъюнкций, тогда выбрасываем все неотмеченные конъюнкции и процесс заканчивается. Оставшееся множество конъюнкций и дает искомую компоненту связности.

Мы видим, что сначала идет изучение покрытия при помощи окрестностей 1-го порядка, и на основе него делаются отметки конъюнкций. Последнее означает, что каждая конъюнкция снабжается одной ячейкой памяти с двумя состояниями (отметки «нет» и отметка «есть»). Кроме того, нужна еще одна ячейка для выяснения, увеличивает ли очередной просмотр число отметок или нет.

В качестве второго параметра возьмем число v — число допустимых ячеек двоичной памяти для каждой конъюнкции. Фиксируем параметр v.

Теперь перейдем к описанию локальных алгоритмов *) [6] над покрытиями максимальными гранями с параметрами u и v. Работа алгоритма разбивается на два этапа.

*) Здесь мы даем несколько другой вариант этого понятия.

I. Изучение покрытия. В определенном порядке просматривают сокращенную д. н. ф. и на основе того, что попадает в окрестность $S_u(N_{K^0})$ конъюнкции K^0, т. е. части сокращенной д. н. ф. и информации, накопленной для конъюнкций из этой окрестности, вычисляют новые значения $\gamma_1^0, \ldots, \gamma_v^0$ ячеек памяти для K^0. При этом требуется, чтобы это вычисление выполнялось одинаково для любых двух д. н. ф., имеющих конъюнкцию K^0 и таких, что у них окрестности конъюнкции K^0 совпадают и имеют одинаковые значения ячеек памяти для конъюнкций из данной окрестности (локальность накопления информации). При выполнении определенных требований процесс накопления информации оказывается «сходящимся» и тем самым может быть окончен. В этих случаях мы получаем финальную информацию, задаваемую значениями ячеек памяти для конъюнкций из сокращенной д. н. ф.

II. Принятие решения. По вычисленным значениям ячеек памяти конъюнкций из $S_u(N_{K^0})$ определяют возможность удаления конъюнкции K^0. Эта процедура также осуществляется локальным образом, т. е. одинакова для любых двух д. н. ф., содержащих K^0, у которых окрестности K^0 совпадают и имеют одинаковые значения ячеек памяти для конъюнкций из этой окрестности.

Легко видеть, что сформулированный выше алгоритм для нахождения в сокращенной д. н. ф. компоненты связности, содержащей данную конъюнкцию K^0, является локальным алгоритмом с параметрами $u = 1$, $v = 1$.

В дальнейшем будем предполагать, что f такова, что покрытие множества N_f совокупностью всех ее максимальных граней образует связную компоненту. В противном случае задача минимизации с использованием локальных алгоритмов решается независимо для каждой связной компоненты в отдельности.

Пусть $f(x_1, \ldots, x_n)$ обладает указанным свойством. Нетрудно видеть, что существует локальный алгоритм с параметрами $u = 1$, $v = 2^n$, который позволяет строить минимальную д. н. ф.

I этап. Сопоставим взаимно однозначным образом наборы $(\alpha_1, \ldots, \alpha_n)$ с числами из множества $\{1, 2, \ldots, 2^n\}$:

$$(\alpha_1, \ldots, \alpha_n) \leftrightarrow i(\alpha_1, \ldots, \alpha_n).$$

Каждой конъюнкции K^0 из сокращенной д. н. ф. для f

сопоставляем двоичный кортеж

$$(\gamma^0_1, \ldots, \gamma^0_{2^n}),$$

где $\gamma^0_{i(\alpha_1,\ldots,\alpha_n)} = 1$ тогда и только тогда, когда $K^0(\alpha_1, \ldots, \alpha_n) = 1$. Очевидно, $(\gamma^0_1, \ldots, \gamma^0_{2^n})$ будет «кодом» грани K^0. Затем идет изучение покрытия при помощи окрестностей 1-го порядка, и новые значения $(\gamma^0_1, \ldots, \gamma^0_{2^n})$ для K^0 вычисляются как покомпонентные дизъюнкции кортежей для данной окрестности. Этот процесс приведет к тому, что для каждой конъюнкции сокращенной д. н. ф. мы вычислим двоичный кортеж и он будет одним и тем же для всех конъюнкций — «кодом» функции (например, столбцом, определяющим ее в таблице).

II этап. Решающее правило определим так: возьмем произвольную минимальную д. н. ф. для f и данную конъюнкцию выбрасываем тогда и только тогда, когда она не входит в минимальную д. н. ф. Очевидно, данное решающее правило удовлетворяет требованию локальности и приводит нас к минимальной д. н. ф.

Данное обстоятельство означает, что в локальных алгоритмах необходимо вводить ограничения на объем допустимой памяти v.

В заключение этого параграфа покажем, что алгоритм Квайна и алгоритм построения д. н. ф. типа ΣT являются локальными и произведем оценку их параметров.

В алгоритме Квайна сначала выявляются ядровые конъюнкции. Для этого необходимо знать окрестности первого порядка конъюнкций в сокращенной д. н. ф., и запоминать отметки: если конъюнкция не ядровая, то отметка 0, если — ядровая, то отметка 1. Для принятия решения о возможности удаления конъюнкции надо опять знать ее окрестность первого порядка и посмотреть, покрывается ли она отмеченными конъюнкциями из этой окрестности. Таким образом, мы имеем локальный алгоритм с параметрами $u = 1$, $v = 1$.

В алгоритме построения д. н. ф. типа ΣT сначала определяют, является ли данная конъюнкция регулярной. Для этого нужно сравнивать пучки, проходящие через точки данной грани (конъюнкции) и пучки, проходящие через точки из окрестности 1-го порядка данной грани и не лежащие в ней, т. е. оперировать с окрестностями 2-го порядка. Регулярная грань помечается символом 1,

грань, не являющаяся регулярной,— 0. Затем происходит принятие решения: удаляются грани с пометкой 1 (регулярные). Итак, в этом случае мы имеем локальный алгоритм с параметрами $u = 2$ и $v = 1$.

На основании обсуждений мы видим, что локальные алгоритмы охватывают многие известные классы алгоритмов. С другой стороны, локальные алгоритмы с параметрами u и v являются алгоритмами с ограниченной трудоемкостью.

Глава 2

СИНТЕЗ СХЕМ ИЗ ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ

В современной технике управляющих и вычислительных устройств важное место занимают *дискретные преобразователи*, т. е. устройства (см. рис. 1), которые обладают некоторым числом входов и выходов. Наборы сигналов, поступающие на входы и возникающие на выходах, принадлежат известным конечным множествам. Устройства осуществляют преобразования входных наборов сигналов в выходные. Интересным подклассом дискретных преобразователей является класс устройств, в которых время преобразования существенно мало́ по сравнению с длительностью сигналов (или устройства, временем преобразования в которых можно пренебречь). Математической моделью таких устройств являются так называемые схемы из функциональных элементов.

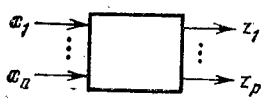


Рис. 1

§ 1. Понятие схемы из функциональных элементов

Определение понятия схемы из функциональных элементов (Ф. Э.) можно разбить на два этапа. На первом этапе раскрывается структурная (схемная) часть этого понятия, на втором — функциональная.

I этап. Определение схемы из функциональных элементов с точки зрения ее структуры. Этот этап, в свою очередь, разбивается на ряд пунктов.

1°. Имеется конечное множество F объектов F_i $(i = 1, \ldots, r)$, называемых *элементами*. Каждый эле-

мент F_i имеет n_i входов и один выход. Элемент F_i графически изображается так, как указано на рис. 2.

2°. Исходя из элементов, по индукции определяем понятие логической сети (геометрическое определение). *Логическая сеть* Σ будет определяться как объект (см. рис. 3), в котором имеется некоторое число n входов и некоторое число p выходов.

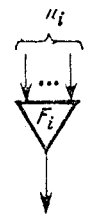


Рис. 2

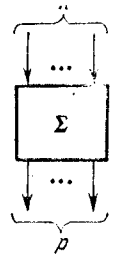


Рис. 3

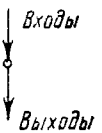


Рис. 4

а) Базис индукции. Одна изолированная вершина называется (*тривиальной*) *логической сетью*. По определению, она является одновременно входом и выходом (см. рис. 4). Здесь и далее на рисунках входящая стрелка обозначает вход, исходящая стрелка — выход.

б) Индуктивный переход. Эта часть основана на использовании трех операций.

I°. Операция объединения непересекающихся сетей. Пусть Σ' и Σ'' — две непересекающиеся

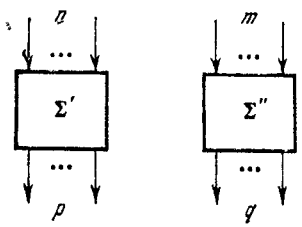


Рис. 5

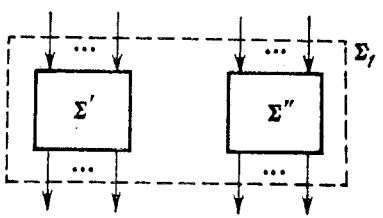


Рис. 6

сети (без общих элементов, входов и выходов), имеющие соответственно n и m входов и p и q выходов (см. рис. 5). Теоретико-множественное объединение двух непересекающихся логических сетей Σ' и Σ'' есть логическая сеть Σ_1, входами которой являются все входы сетей Σ' и Σ'', выходами — все выходы сетей Σ' и Σ''. Сеть Σ_1 имеет $n+m$ входов и $p+q$ выходов (см. рис. 6).

II°. Операция присоединения элемента F_i. Пусть логическая сеть Σ' и элемент F_i (рис. 7) таковы, что $n_i \leqslant p$ и в Σ' выбрано n_i попарно различных выходов

с номерами $j_1, j_2, \ldots, j_{n_i}$. Тогда фигура Σ_2 называется логической сетью, являющейся результатом подключения элемента F_i к логической сети Σ'. Входами Σ_2 являются все входы Σ', выходами — все выходы сети Σ', кроме выходов с номерами $j_1, \ldots, j_{n_i}$, а также выход элемента F_i. Логическая сеть Σ_2 имеет n входов и $p - n_i + 1$ выходов.

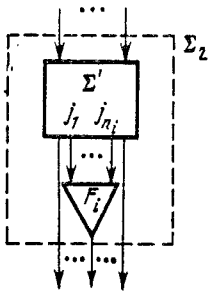


Рис. 7

III°. Операция расщепления выхода. Пусть в логической сети Σ' (рис. 8) выделен выход с номером j. Тогда фигура Σ_3 называется логической сетью, полученной путем расщепления выхода j. Входами Σ_3 являются все входы Σ', выходами — все выходы логической сети Σ' с номерами $1, \ldots, j-1, j+1, \ldots, p$ и еще два выхода, возникших из выхода с номером j сети Σ'. Следовательно, Σ_3 имеет n входов и $p+1$ выход.

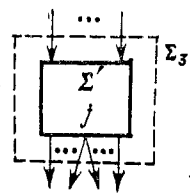


Рис. 8

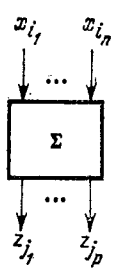


Рис. 9

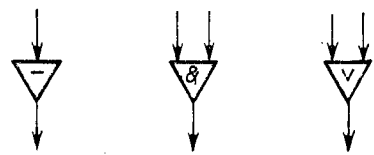


Рис. 10

3°. Пусть заданы алфавиты переменных $X = \{x_i\}$ и $Z = \{z_i\}$ *). Рассмотрим логическую сеть Σ, имеющую n входов и p выходов.

Схемой из функциональных элементов называется логическая сеть, входам и выходам которой приписаны раз-

*) Здесь символ $\{x_i\}$ обозначает множество всех x_i, где индекс i пробегает натуральный ряд (или его подмножество).

личные буквы $x_{i_1}, \ldots, x_{i_n}$ и $z_{j_1}, \ldots, z_{j_p}$ соответственно из алфавитов X и Z (см. рис. 9). Полученную таким образом схему будем обозначать через

$$\Sigma\left(x_{i_1}, \ldots, x_{i_n}; z_{j_1}, \ldots, z_{j_p}\right).$$

Приведем примеры схем.

1. Пусть множество F состоит из трех элементов (см. рис. 10).

Тогда фигура Σ_1, изображенная на рис. 11, будет схемой, так как она может быть построена с использовани-

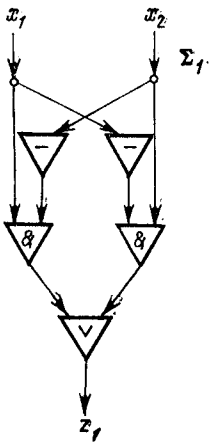


Рис. 11

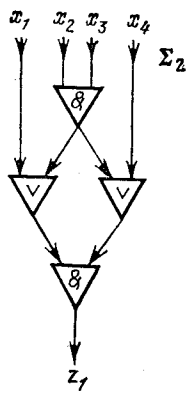


Рис. 12

ем F при помощи операций I°—III°. Этапы этого построения изображены в табл. 1.

2. Фигура Σ_2, изображенная на рис. 12, будет также схемой (см. табл. 2).

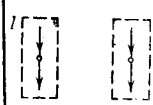
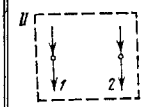
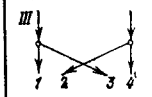
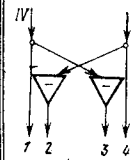
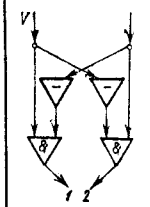
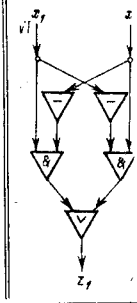
В дальнейшем мы встретимся с примерами схем, у которых вход является одновременно и выходом и у которых возможно несколько выходов.

II этап. Определение функционирования схемы.

4°. Пусть $\Sigma(x_1, \ldots, x_n; z_1, \ldots, z_p)$ — схема из функциональных элементов*). Сопоставим ей систему уравнений

*) Без ограничения общности можно считать, что номера переменных образуют начальные отрезки натурального ряда.

Таблица 1

Логическая сеть	Способ получения	Логическая сеть	Способ получения
	Берутся две тривиальные логические сети (Базис индукции)		К I применяется операция I
	В II производят разветвление выходов 1 и 2. Операция III (2 раза)		К выходам 2 и 3 сети III подключают элементы $\bar{}$ Операция II (2 раза)
	К выходам (1, 2) и (3, 4) сети IV подключают элементы &. Операция II (2 раза)		К выходам 1, 2 сети V подключают элемент ∨. Операция II

(функций) алгебры логики

$$z_1 = f_1(x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots$$
$$z_p = f_p(x_1, \ldots, x_n),$$

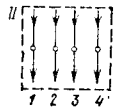
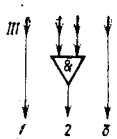
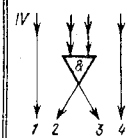
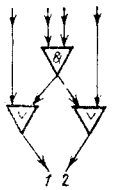
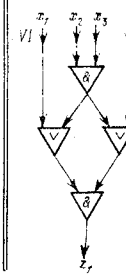
называемую также *проводимостью данной схемы*. Для этого каждому элементу F_i из множества F ставится в соответствие логический оператор $f_i^0(\ldots, \ldots, \ldots)$, имеющий n_i мест и задаваемый булевой функцией $f_i^0(y_1, \ldots, y_{n_i})$. Далее по индукции определяется проводимость схемы.

а) Базис индукции. Схема Σ — тривиальная схема (см. рис. 13). В этом случае уравнение имеет вид

$$z_1 = x_1,$$

и проводимость есть тождественная функция.

Таблица 2

Логическая сеть	Способ получения	Логическая сеть	Способ получения
	Берутся четыре тривиальные логические сети		К I применяется операция I (3 раза)
	К выходам (2, 3) сети II подключают элемент &. Операция II		В III производят разветвление выхода 2. Операция III
	К выходам (1, 2) и (3, 4) сети IV подключают элементы $\vee$. Операция II (2 раза)		К выходам (1, 2) сети V подключают элемент &. Операция II

б) Индуктивный переход. Пусть Σ' и Σ'' — две схемы, сети которых не пересекаются и входам и

Рис. 13

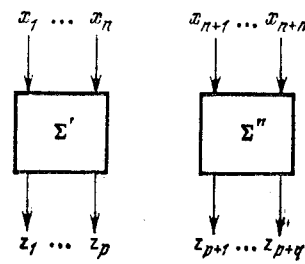


Рис. 14

выходам которых приписаны, соответственно, различные буквы (см. рис. 14). Пусть также схемам

$\Sigma'(x_1, \ldots, x_n; z_1, \ldots, z_p)$, $\Sigma''(x_{n+1}, \ldots, x_{n+m}; z_{p+1}, \ldots, z_{p+q})$

соответствуют системы уравнений

$$\begin{array}{l} z_1 = f_1(x_1, \ldots, x_n), \\ \ldots\ldots\ldots \\ z_p = f_p(x_1, \ldots, x_n), \end{array} \quad (*) \qquad \begin{array}{l} z_{p+1} = f_{p+1}(x_{n+1}, \ldots, x_{n+m}), \\ \ldots\ldots\ldots\ldots \\ z_{p+q} = f_{p+q}(x_{n+1}, \ldots, x_{n+m}). \end{array} \quad (**)$$

I. Схеме $\Sigma_1(x_1, \ldots, x_{n+m};\ z_1, \ldots, z_{p+q})$, являющейся объединением двух данных схем, поставим в соответствие систему уравнений, представляющую собой объединение уравнений (*) и (**):

$$\begin{array}{l} z_1 = f'_1(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}), \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ z_p = f'_p(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}), \\ z_{p+1} = f'_{p+1}(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}), \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ z_{p+q} = f'_{p+q}(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}). \end{array}$$

Здесь $f'_1, \ldots, f'_p$ не зависят существенно от переменных $x_{n+1}, \ldots, x_{n+m}$, а $f'_{p+1}, \ldots, f'_{p+q}$ — от переменных $x_1, \ldots, x_n$ и

$$f'_1 = f_1, \ \ldots, f'_p = f_p, \quad f'_{p+1} = f_{p+1}, \ \ldots, f'_{p+q} = f_{p+q}.$$

II. Пусть схема

$$\Sigma_2(x_1, \ldots, x_n;\ z_1, \ldots, z_{j_1-1}, z_{j_1+1}, \ldots, z_{j_{n_i}-1}, z_{j_{n_i}+1}, \ldots, z_p, z_{p+1})$$

получена из схемы

$$\Sigma'(x_1, \ldots, x_n;\ z_1, \ldots, z_p)$$

путем присоединения к выходам $z_{j_1}, \ldots, z_{j_{n_i}}$ $(n_i \leqslant p)$ элемента F_i, и полученному новому выходу приписана буква z_{p+1}. Тогда сопоставим этой схеме систему уравнений, получающуюся из (*) вычеркиванием $j_1, \ldots, j_{n_i}$ строчек с добавлением строки

$$z_{p+1} = f_i^0\left(f_{j_1}(x_1, \ldots, x_n), \ldots, f_{j_{n_i}}(x_1, \ldots, x_n)\right).$$

Получим

$$
\begin{gathered}
z_1 = f_1(x_1, \ldots, x_n),\\
\cdots\cdots\cdots\cdots\\
z_{j_1-1} = f_{j_1-1}(x_1, \ldots, x_n),\\
z_{j_1+1} = f_{j_1+1}(x_1, \ldots, x_n),\\
\cdots\cdots\cdots\cdots\\
z_{j_{n_i}-1} = f_{j_{n_i}-1}(x_1, \ldots, x_n),\\
z_{j_{n_i}+1} = f_{j_{n_i}+1}(x_1, \ldots, x_n),\\
\cdots\cdots\cdots\cdots\\
z_p = f_p(x_1, \ldots, x_n),\\
z_{p+1} = f_i^0\left(f_{j_1}(x_1, \ldots, x_n), \ldots, f_{j_n}(x_1, \ldots, x_n)\right).
\end{gathered}
$$

III. Пусть схема $\Sigma_3(x_1, \ldots, x_n; z_1, \ldots, z_p, z_{p+1})$ получена из схемы $\Sigma'(x_1, \ldots, x_n; z_1, \ldots, z_p)$ расщеплением выхода с номером j на два выхода, и полученным выходам приписаны соответственно буквы z_j и z_{p+1}. Тогда этой схеме поставим в соответствие систему уравнений

$$
\begin{gathered}
z_1 = f_1(x_1, \ldots, x_n),\\
\cdots\cdots\cdots\\
z_{j-1} = f_{j-1}(x_1, \ldots, x_n),\\
z_j = f_j(x_1, \ldots, x_n),\\
z_{j+1} = f_{j+1}(x_1, \ldots, x_n),\\
\cdots\cdots\cdots\\
z_p = f_p(x_1, \ldots, x_n),\\
z_{p+1} = f_j(x_1, \ldots, x_n).
\end{gathered}
$$

Таким образом, каждой схеме из Ф. Э. сопоставлена система уравнений, которая по существу определяется системой булевых функций, входящих в их правую часть.

Пример 1. а) Для схемы Σ_1 (рис. 11) имеем систему, состоящую из одного уравнения

$$z_1 = x_1 \& \bar{x}_2 \vee \bar{x}_1 \& x_2 \text{ или } z_1 = x_1 + x_2 \pmod 2.$$

б) Для схемы Σ_2 (рис. 12) аналогично получаем

$$z_1 = x_1 x_4 \vee x_2 x_3.$$

Замечание. Схему из Ф. Э. $\Sigma(x_1, \ldots, x_n; z_1, \ldots, z_p)$ можно рассматривать как соединение элементов F_i. Каждый элемент F_i является элементарным преобразователем, который преобразует входной набор $(\sigma_1, \ldots, \sigma_{n_i})$ в выходное значение $f_i^0(\sigma_1, \ldots, \sigma_{n_i})$. В таком случае, если на входы схемы Σ подать набор $(\alpha_1, \ldots, \alpha_n)$, то, продвигаясь

от входов к выходам, он будет переходить в набор $(\gamma_1, \ldots, \gamma_p)$, характеризующий состояния выходов. Оказывается, что

$$\gamma_1 = f_1(\alpha_1, \ldots, \alpha_n),$$
$$\ldots\ldots\ldots\ldots$$
$$\gamma_p = f_p(\alpha_1, \ldots, \alpha_n),$$

т. е. преобразование, описываемое данными уравнениями, соответствует нашему интуитивному представлению о функционировании схемы из Ф. Э.

Обозначим через $\mathfrak{S}$ класс всех схем из Ф. Э. над F. Пусть $\mathfrak{S}_0$ — подкласс всех схем из Ф. Э. над F, у которых:

1) имеется ровно один выход;

2) разветвления имеются только на входах.

Очевидно, что $\Sigma_1 \in \mathfrak{S}_0$, а $\Sigma_2 \notin \mathfrak{S}_0$ над базисом F (см. рис. 11, 12). Обозначим $\mathfrak{S}_{\mathfrak{P}}$ класс формул над системой $\mathfrak{P} = \{f_i^0(x_1, \ldots, x_{n_i})\}$.

Оказывается, что между классами $\mathfrak{S}_0$ и $\mathfrak{S}_{\mathfrak{P}}$ существует вполне определенная связь, которая может быть математически строго сформулирована. Более грубо эта связь выражается следующим образом: каждой схеме Σ из $\mathfrak{S}_0$ можно однозначным образом сопоставить формулу $\mathfrak{A}$ из $\mathfrak{S}_{\mathfrak{P}}$ так, что по формуле $\mathfrak{A}$ исходная схема восстанавливается в некотором смысле однозначно с точностью до символа, приписанного выходу.

Данное утверждение можно проиллюстрировать на примере. Рассмотрим схему Σ_1 (см. рис. 11). Очевидно, что $\Sigma_1 \in \mathfrak{S}_0$. Этой схеме соответствует формула $\mathfrak{A}$, где

$$\mathfrak{A} = (x_1 \,\&\, \bar{x}_2) \vee (\bar{x}_1 \,\&\, x_2)$$

(см. процесс построения соответствующих уравнений). Очевидно, что эта формула позволяет восстановить структуру исходной схемы Σ.

Доказательство утверждения несложно. Для этого нужно сравнить определение схемы из $\mathfrak{S}_0$ и формулы над соответствующей системой.

В силу сказанного схемы из $\mathfrak{S}_0$ можно рассматривать как формулы или, точнее, как геометрические аналоги формул.

Заметим, что если схема Σ содержит элемент F_i, который на выходе имеет разветвление, то его можно заменить на группу элементов F_i, у которых на выходах отсутствуют разветвления (см. рис. 15), и при этом функционирование схемы не изменится. Эта замена по-

зволяет преобразовать схему Σ в схему Σ', функционирование которой описывается теми же уравнениями, но имеет разветвления только на входах. Таким образом, функциональные возможности класса $\mathfrak{S}$ схем Σ над заданным базисом совпадают с функциональными возможностями подкласса его всех схем, у которых разветвления могут быть только на входах. Каждая схема Σ' из указанного подкласса представляет собой «склейку» схем из $\mathfrak{S}_0$ своими входами. Отсюда функциональные возможности класса $\mathfrak{S}$ определяются функциональными возможностями подкласса $\mathfrak{S}_0$, т. е. возможностями $\mathfrak{S}_{\mathfrak{P}}$. Мы доказали следующую теорему.

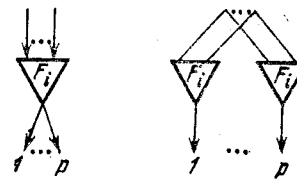


Рис. 15

Теорема 1 (о полноте). *Для того чтобы для произвольной системы булевых уравнений вида*

$$z_1 = f_1(x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots$$
$$z_p = f_p(x_1, \ldots, x_n),$$

существовала схема $\Sigma(x_1, \ldots, x_n; z_1, \ldots, z_p)$ *в исходном базисе Ф. Э., реализующая эту систему уравнений, необходимо и достаточно, чтобы система* $\mathfrak{P}$ *функций* f_i *была полной.*

Мы видим, что связь классов $\mathfrak{S}_0$ и $\mathfrak{S}_{\mathfrak{P}}$ позволяет свести исследование проблемы полноты класса $\mathfrak{S}$ к проблеме полноты для формул.

Дальнейшее изложение связано с рассмотрением классов $\mathfrak{S}$ схем из Ф. Э., для которых система $\mathfrak{P}$ полна.

§ 2. Проблема синтеза схем из Ф. Э.

Проблема синтеза схем из Ф. Э. состоит в следующем. Задан базис F функциональных элементов и взята произвольная система булевых уравнений

$$z_1 = f_1(x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots$$
$$z_p = f_p(x_1, \ldots, x_n).$$

Требуется построить схему $\Sigma(x_1, \ldots, x_n; z_1, \ldots, z_p)$ из данных Ф. Э., реализующую эту систему уравнений.

Как мы видели выше, решение проблемы синтеза существует, если система $\mathfrak{P}$ функций f_i^1 полна. Поскольку уже в случае реализации булевых функций формулами имеется много решений, то для данной системы булевых уравнений (при наличии полноты системы $\mathfrak{P}$) можно построить много схем из Ф. Э., которые реализуют эту систему уравнений.

Ввиду этого задача синтеза требует уточнения, которое позволило бы осуществлять выбор в определенном смысле оптимального решения. Последнее может быть сделано следующим образом. Рассмотрим функционал $L(\Sigma)$, определенный для всех Σ из $\mathfrak{S}$, скажем, как число элементов в схеме Σ. Число $L(\Sigma)$ будем называть *сложностью схемы*. Будем дополнительно требовать, чтобы синтезируемая схема Σ имела минимальную сложность. В дальнейшем такие схемы будем называть *минимальными*. Следовательно, теперь проблема синтеза состоит в построении минимальных схем.

Ближайшая цель изложения состоит в описании алгоритма для построения минимальных схем. В связи с этим введем одно понятие и докажем три леммы.

Пусть задано n входов $x_{i_1}, \ldots, x_{i_n}$, p выходов $z_{j_1}, \ldots, z_{j_p}$ и r элементов $F_1, \ldots, F_r$, которые будут изображаться так, как это указано на рис. 16.

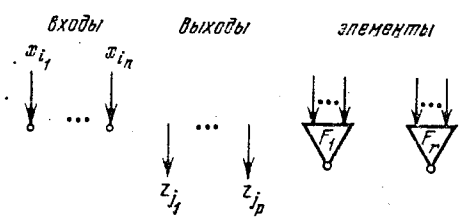


Рис. 16

Определение. *Соединением* S данных входов, выходов и элементов называется геометрическая фигура, состоящая из этих объектов и обладающая следующими свойствами:

1) каждый вход элементов подключен либо ко входу, либо к выходу элемента;

2) каждый выход подключен либо ко входу, либо к выходу элемента.

Замечание. Очевидно, схема $\Sigma(x_{i_1}, \ldots, x_{i_n}, z_{j_1}, \ldots, z_{j_p})$ из Ф. Э. является соединением. В то же время существуют соединения, не являющиеся схемами из Ф. Э. (см. рис. 17).

Лемма 1. *Число* $S^*(n, p, h)$ *соединений с данными входами* $x_{i_1}, \ldots, x_{i_n}$, *данными выходами* $z_{j_1}, \ldots, z_{j_p}$ *и содержащих* h *Ф. Э., занумерованных числами от* 1 *до* h, *не превосходит*

$$r^h (n+h)^{hv+p},$$

где $v = \max\limits_{1 \leqslant i \leqslant r} n_i$.

Доказательство. Каждый из h занумерованных элементов можно выбрать r способами, а каждый из его

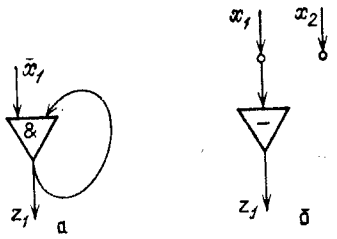


Рис. 17

входов можно подключить либо к одному из входов $x_{i_1}, \ldots, x_{i_n}$, либо к выходу одного из h элементов, т. е. вход элемента может быть подключен $n+h$ способами. Всего для элемента имеется не более $r(n+h)^v$ возможностей. Очевидно, что каждый из выходов $z_{j_1}, \ldots, z_{j_p}$ может быть подключен $n+h$ способами. Поэтому

$$S^*(n, p, h) \leqslant r^h (n+h)^{hv+p}.$$

Теорема 2. *Число* $S_0(n, p, h)$ *минимальных схем из Ф. Э. с данными входами* $x_{i_1}, \ldots, x_{i_n}$, *данными выходами* $z_{j_1}, \ldots, z_{j_p}$ *и содержащих* h *Ф. Э., удовлетворяет неравенству*

$$S_0(n, p, h) \leqslant \frac{1}{h!} r^h (n+h)^{hv+p}.$$

Доказательство. Очевидно, что в минимальной схеме на выходах разных элементов получаются разные функции от переменных $x_{i_1}, \ldots, x_{i_n}$ (иначе один из таких элементов можно было бы удалить из схемы, и, из-

менив некоторые соединения в ней, сохранить ее функционирование). Благодаря этому все $h!$ соединений с занумерованными элементами, которые порождает каждая минимальная схема, различны. Отсюда и из леммы 1 следует неравенство теоремы.

Легко видеть, что определенные выше операции для логических сетей могут быть распространены и на соединения.

I. Операция объединения двух соединений. Применяется к двум соединениям S' и S'', которые не имеют общих входов, выходов и элементов. Результат операции имеет все элементы и все связи обоих соединений. Его входами будут входы S' и S'', выходами — выходы S' и S''.

II. Операция подключения элемента. Берется соединение S' с p выходами и элемент F_i, причем $n_i \leqslant p$. В S' выбираются n_i выходов (приписанные им символы из алфавита Z исключаются) и каким-либо образом подключается элемент F_i; его выходу приписывается символ из алфавита Z, отличный от символов, приписанных другим выходам.

III. Операция разветвления выхода. В соединении S' берется некоторый выход z_{j_t}, который присоединен либо к входу, либо к выходу элемента. Берем новый выход z (символ z отличен от символов, приписанных другим выходам) и подключаем его к той же вершине, что и z_{j_t}.

Лемма 2. *Результат применения операций* I, II, III *к соединениям является соединением.*

Лемма 3. *Соединение S, отличное от тривиальной схемы, является схемой тогда и только тогда, когда выполнено хотя бы одно из условий*:

1) *S получается объединением соединений S' и S'', которые являются схемами*;

2) *S получается из S' путем подключения элемента F_i, и S' является схемой*;

3) *S получается из S' путем разветвления его выхода, и S' является схемой.*

Доказательство этих двух лемм очевидно.

Теорема 3. *Существует алгоритм, который для каждого соединения S выясняет, является ли оно схемой или нет.*

Доказательство ведется индукцией по числу d связей соединения S, т. е. по суммарному числу всех входов элементов и всех выходов S. В случае $d=1$ либо S совпадает с тривиальной схемой, либо не является схемой. Пусть теорема верна для всех $d=1, \ldots, \lambda$. Покажем, что она справедлива и для $d=\lambda+1$. Рассмотрим соединение S, имеющее $\lambda+1$ связь. Очевидно, S отлично от тривиальной схемы ($d \geqslant 2$). Для соединения S возможны два случая:

1) S не получается из соединений с меньшим числом связей путем применения ни одной из операций I—III. Тогда, очевидно, S не является схемой;

2) S является результатом операции над соединениями с меньшим числом связей.

В этом случае просматриваем все варианты разложений (их конечное число) и в силу предположения индукции для компонент разложения выясняем, будут ли они схемами или нет. Если найдется разложение, при котором все компоненты являются схемами, то исходное соединение по предыдущей лемме будет схемой. Если в каждом разложении хотя бы одна из компонент не будет схемой, то исходное соединение не будет схемой. Теорема доказана.

Теорема 4. *Существует алгоритм, который для каждой системы булевых уравнений строит минимальную схему* Σ.

Доказательство. Пусть система уравнений имеет следующий вид:

$$z_1 = f_1(x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$z_p = f_p(x_1, \ldots, x_n).$$

Просматриваем последовательно все соединения со входами $x_1, \ldots, x_n$ и выходами $z_1, \ldots, z_p$ с числом элементов $h=0, 1, 2, \ldots$

Берем $h=0$. Мы имеем конечное число соединений, не содержащих элементов. Для каждого соединения проверяем, является ли оно схемой или нет. Если соединение есть схема, то выясняем, реализует ли оно данную систему уравнений. Таким образом, либо мы найдем требуемую схему (она, очевидно, будет минимальной) или среди схем сложности $h=0$ искомой схемы нет. В последнем случае минимальная схема имеет сложность $L(\Sigma)>0$.

Берем $h = 1$. Мы имеем конечное число соединений, содержащих один элемент. Для каждого соединения проверяем, будет ли оно схемой или нет. В случае, когда соединение есть схема, выясняем, реализует ли оно данную систему уравнений. В результате этого мы либо найдем требуемую схему, и она будет минимальной, либо среди схем сложностей $h = 0$ и $h = 1$ искомой схемы нет. Значит, минимальная схема Σ имеет сложность $L(\Sigma) > 1$ и т. д. В силу полноты системы $\mathfrak{P}$, этот процесс должен окончиться, и в результате мы построим минимальную схему. Теорема доказана.

Приведенный выше алгоритм для построения минимальных схем относится к классу алгоритмов типа «полного перебора», так как он основан на просмотре всех схем до определенной сложности. Алгоритмы полного перебора, как правило, обладают большой трудоемкостью. Они, по существу, не дают возможности изучать свойства искомых объектов и непригодны для практических целей. Ввиду этого далее рассматривается более простая задача, для которой исходная система уравнений содержит одно уравнение

$$z = f(x_1, \ldots, x_n),$$

и, следовательно, искомая схема Σ имеет один выход.

Сложность минимальной схемы будем обозначать через $L(f)$.

Кроме того, вместо синтеза схем для отдельных функций (уравнений) рассматривается задача синтеза для класса $P^{(n)}$ всех функций от n переменных. При этом качество алгоритмов синтеза сравнивается путем сопоставления так называемых функций Шеннона. Более точно, пусть

$$L(n) = \max_{f \in P^{(n)}} L(f), \quad L_A(n) = \max_{f \in P^{(n)}} L_A(f),$$

где $L_A(f)$ — минимальная сложность схем, реализующих f, которые получаются при помощи алгоритма A.

Функции $L(n)$, $L_A(n)$ называются *функциями Шеннона*. Они, очевидно, характеризуют сложность класса $P^{(n)}$ с точки зрения реализаций его функций минимальными схемами, соответственно минимальными относительно данного алгоритма схемами, и

$$L_A(n) \geqslant L(n).$$

Задача синтеза состоит теперь в том, чтобы найти алгоритм A, для которого $L_A(n)$ была бы возможно ближе к $L(n)$ (например, $L_A(n) = L(n)$), и чтобы трудоемкость алгоритма A была существенно меньше, чем трудоемкость алгоритма полного перебора. Следует обратить внимание на то, что в новой постановке задачи мы не требуем, чтобы алгоритм A для каждой функции f находил минимальную схему, необходимо только, чтобы простейшая схема, получаемая при помощи A, имела сложность $L_A(f)$, сильно не превышающую $L(n)$.

§ 3. Элементарные методы синтеза

Мы приведем несколько алгоритмов синтеза, использующих элементарные идеи, в случае когда базис F состоит из инвертора, дизъюнктора и конъюнктора.

а) Метод синтеза, основанный на совершенной д. н. ф.

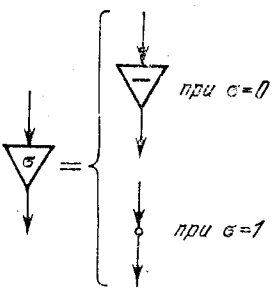


Рис. 18

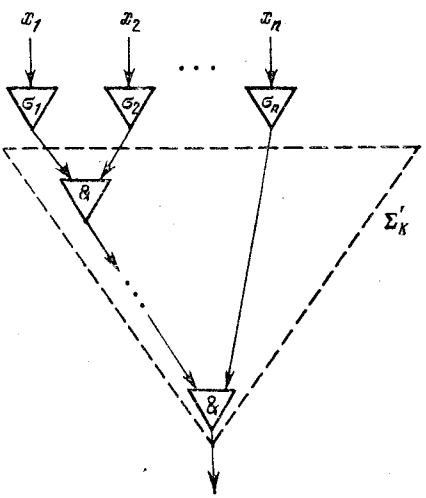


Рис. 19

Рассмотрим разложение функции $f(x_1, \ldots, x_n)$ ($f \not\equiv \text{const}$) в виде совершенной д. н. ф.:

$$f(x_1, \ldots, x_n) = \bigvee_{\substack{(\sigma_1 \ldots \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n} = \bigvee_{i=1}^{s} K_i.$$

Введем вспомогательный «элемент» (см. рис. 18), с помощью которого легко изобразить (см. рис. 19) схему Σ_K,

реализующую конъюнкцию K, где

$$K = x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Очевидно, $L(\Sigma_K) \leqslant n + (n-1)$, и Σ_K содержит подсхему Σ'_K, одинаковую для всех конъюнкций и имеющую сложность $n-1$. Если «склеить» схемы $\Sigma_{K_1}, \ldots, \Sigma_{K_s}$, начиная от входов $x_1, \ldots, x_n$ вплоть до вспомогательных элементов, то получим схему Σ (см. рис. 20).

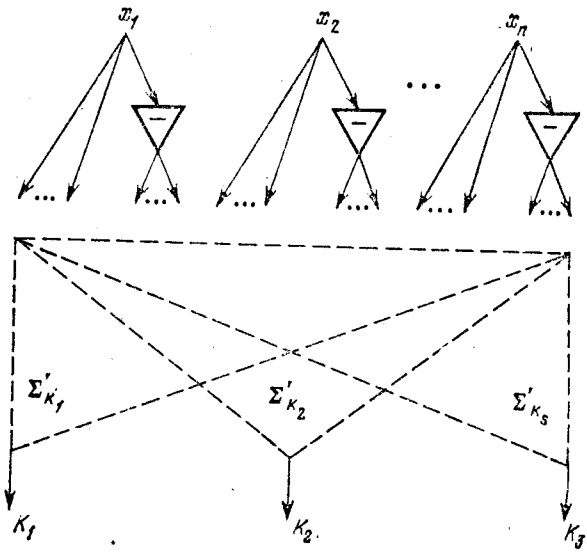


Рис. 20

Мы имеем $L(\Sigma) \leqslant n + s(n-1)$. Подключая выходы схемы Σ к схеме из дизъюнкторов, мы получим схему для $f(x_1, \ldots, x_n)$ (см. рис. 21).

Этим завершено описание метода синтеза по совершенной д. н. ф. (алгоритм A_1). Окончательно получаем

$$L_{A_1}(f) \leqslant n + s(n-1) + s - 1 < n + ns = n(s+1).$$

Поскольку $f \not\equiv 1$, то $s \leqslant 2^n - 1$ и $L_{A_1}(f) \leqslant n2^n$, а также

$$L_{A_1}(n) \leqslant n2^n.$$

б) Метод синтеза, основанный на более компактной реализации множества всех конъюнкций $\{x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}\}$.

На рис. 22 представлено индуктивное построение многополюсника Σ^n $(n = 1, 2, \ldots)$, реализующего множество всех конъюнкций $\{x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}\}$. Мы имеем

$$L(\Sigma^1) = 1,$$
$$L(\Sigma^n) = L(\Sigma^{n-1}) + 1 + 2^n,$$
$$L(\Sigma^n) = 2^2 + \ldots + 2^n + n = 2 \cdot 2^n + n - 4.$$

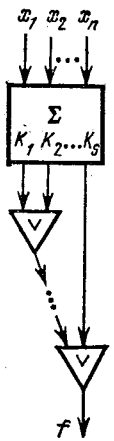


Рис. 21

Для построения схемы, реализующей функцию $f(x_1, \ldots, x_n)$, нужно в многополюснике Σ^n отобрать выходы, соответствующие членам ее совершенной д. н. ф. $K_1, \ldots, K_s$, подключить их к схеме (см. рис. 21), осуществляющей логическое сложение, и удалить лишние элементы. Это потребует еще не более

$$s \leqslant 2^n - 1$$

элементов $\vee$.

Таким образом, этот метод (алгоритм A_2) дает

$$L_{A_2}(f) \leqslant 3 \cdot 2^n + n - 5 \text{ и } L_{A_2}(n) \leqslant 3 \cdot 2^n + n - 5.$$

в) Метод синтеза, основанный на разложении функции $f(x_1, \ldots, x_n)$ по переменному x_n.

Возьмем разложение

$$f(x_1, \ldots, x_{n-1}, x_n) =$$
$$= x_n \& f(x_1, \ldots, x_{n-1}, 1) \vee \bar{x}_n \& f(x_1, \ldots, x_{n-1}, 0)$$

и для краткости положим

$$f' = f(x_1, \ldots, x_{n-1}, 1), \quad f'' = f(x_1, \ldots, x_{n-1}, 0).$$

На рис. 23 представлена индуктивная процедура построения схемы для f.

На основе этого метода имеем алгоритм A_3:

$$L_{A_3}(1) = 2,$$

$$L_{A_3}(n) \leqslant 2L_{A_3}(n-1) + 4.$$

Окончательно имеем

$$L_{A_3}(n) \leqslant 3 \cdot 2^n - 4.$$

Если учесть информацию о реализации функций от двух переменных, например, что $L_{A_3}(2) \leqslant 5$, то можно

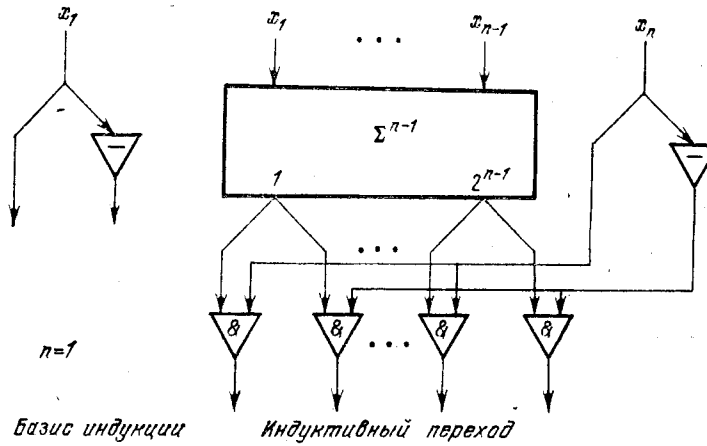


Рис. 22

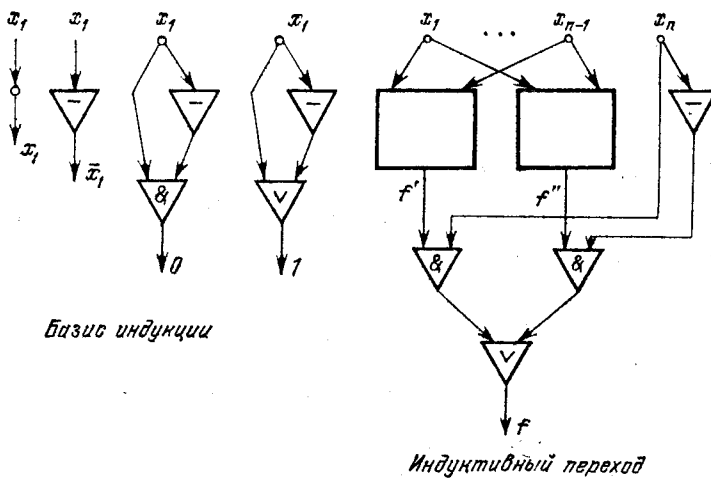


Рис. 23

получить лучшую оценку:

$$L_{A_3}(n) \leqslant 2{,}25 \cdot 2^n - 4.$$

Итак, мы видим, что построенные алгоритмы A_1, A_2 и A_3 в некотором смысле дают возможность получить все более компактные реализации для функций и, в конечном счете, все более хорошие оценки для функций Шеннона. С другой стороны, получение более хороших результатов синтеза достигается за счет некоторого усложнения алгоритма.

§ 4. Нижняя оценка для $L(n)$

Для того чтобы можно было судить о качестве алгоритмов синтеза, нужно знать, насколько отличается величина $L_A(f)$ от $L(f)$. Это сравнение осуществить невозможно, так как величина $L_A(f)$ вычисляется довольно сложно (например, для алгоритмов A_2 и A_3), а $L(f)$ хотя и вычислима, но для большинства функций с практически недоступной сложностью.

Ввиду этого качество алгоритма синтеза оценивают путем сопоставления функций Шеннона $L_A(n)$ и $L(n)$. К сожалению, сравнивать величины $L_A(n)$ и $L(n)$ для каждого n мы не можем, так как, хотя $L_A(n)$ и вычисляется легче, чем $L_A(f)$, функция $L(n)$ вычисляется с практически недоступной сложностью. Поэтому приходится ограничиваться асимптотическим сравнением $L_A(n)$ и $L(n)$, т. е. их сравнением при $n \to \infty$.

Напомним следующие понятия из анализа. Пусть $\varphi(n)$ и $\psi(n)$ — вещественные положительные функции от натуральной переменной n.

1. *Функция* $\varphi(n)$ *асимптотически больше или равна функции* $\psi(n)$, *т. е.* $\varphi(n) \succcurlyeq \psi(n)$, если для любого $\varepsilon > 0$ найдется $N = N(\varepsilon)$ такое, что при любом $n \geqslant N$ $\varphi(n) \geqslant (1-\varepsilon)\psi(n)$.

2. В случае, если $\varphi(n) \succcurlyeq \psi(n)$ и $\psi(n) \succcurlyeq \varphi(n)$, говорят, что $\varphi(n)$ и $\psi(n)$ *асимптотически равны* (*эквивалентны*) и пишут $\varphi(n) \sim \psi(n)$. Здесь, очевидно, предел $\varphi(n)/\psi(n)$ существует и равен 1.

3. Если $0 < c_1 < \varphi(n)/\psi(n) < c_2$, то говорят, что функции $\varphi(n)$ и $\psi(n)$ *эквивалентны с точностью до порядка*. В этом случае пишут $\varphi(n) \asymp \psi(n)$.

Для сравнения асимптотики $L_A(n)$ и $L(n)$ важно получить достаточно хорошую нижнюю асимптотическую оценку для $L(n)$ для базиса, состоящего из инвертора, конъюнктора и дизъюнктора.

Теорема 5. $L(n) \geqslant 2^n/n$.

Доказательство. Как мы видели выше (теорема 2), число минимальных схем из Ф. Э. с n входами и одним выходом ($p = 1$), содержащих ровно h элементов в рассматриваемом базисе ($v = 2$, $r = 3$), т. е. $S_0(n, 1, h)$, оценивается следующим образом:

$$S_0(n,\ 1,\ h) \leqslant \frac{1}{h!}\, 3^h (n+h)^{2h+1}.$$

Обозначим число минимальных схем из Ф. Э. с данными входами $x_1, \ldots, x_n$, данными выходами $z_1, \ldots, z_p$, содержащих не более h элементов, через $S(n, p, h)$. Тогда

$$S(n, p, h) = \sum_{i=0}^{h} S_0(n, p, i).$$

Оценим сверху $S(n, 1, h)$. Мы имеем

$$S(n, 1, h) = \sum_{i=0}^{h} S_0(n, 1, i) \leqslant \sum_{i=0}^{h} \frac{1}{i!}\, 3^i (n+i)^{2i+1} \leqslant$$

$$\leqslant (h+1)\, \frac{1}{h!}\, 3^h (n+h)^{2h+1} \leqslant \frac{1}{h!}\, 3^h (n+h)^{2h+2}.$$

Если $h > n$, то

$$S(n, 1, h) < \frac{3^h (2h)^{2h+2}}{(h/e)^h} = 4h^2 (12e)^h h^h < (ch)^h,$$

где c — некоторая константа.

Положим $h = [2^n/n]$ (при этом условие $h > n$ выполняется, начиная с $n = 5$) и оценим сверху число минимальных схем из Ф. Э. сложности не более h (при $n \geqslant 5$). Мы имеем

$$S(n, 1, h) \leqslant \left(c\, \frac{2^n}{n}\right)^{2^n/n},$$

и, значит, минимальными схемами сложности не более h может быть реализовано не более $\left(c\, \frac{2^n}{n}\right)^{2^n/n}$ булевых функций от n переменных.

Рассмотрим выражение

$$\log_2 \frac{\left(c\frac{2^n}{n}\right)^{2^n/n}}{2^{2^n}} = \frac{2^n}{n}(\log_2 c + n - \log_2 n) - 2^n =$$

$$= \frac{2^n}{n}(\log_2 c - \log_2 n) \to -\infty \quad (\text{при } n \to \infty).$$

Следовательно, при достаточно большом n числитель будет меньше знаменателя. Значит, минимальных схем сложности не более h не хватает для реализации всех булевых функций от n переменных, и поэтому существуют функции от n переменных, которые не могут быть реализованы со сложностью, меньшей или равной

$$h = \left[\frac{2^n}{n}\right],$$

т. е. при достаточно больших n

$$L(n) > \frac{2^n}{n}.$$

Теорема доказана.

Следствие. *Доля тех функций f, для которых $L(f) > 2^n/n$, стремится к единице при $n \to \infty$.*

§ 5. Оптимальный по порядку метод синтеза схем из Ф. Э. (метод Шеннона)

Полученная выше нижняя оценка для $L(n)$ показывает, что наилучшие из элементарных методов синтеза отличаются по порядку от нижней оценки в n раз. Это означает, что дальнейшее совершенствование методов синтеза может привести к уменьшению верхней оценки для функции Шеннона по порядку не более чем в n раз по сравнению с $c2^n$. В действительности оказалось, что существует такой метод синтеза, который приводит к верхней оценке для функции Шеннона, по порядку совпадающей с нижней оценкой. Этот метод был создан К. Шенноном для контактных схем. Здесь мы изложим этот метод для реализации булевых функций схемами из Ф. Э.

Пусть $\{f_1(x_1, \ldots, x_n), \ldots, f_s(x_1, \ldots, x_n)\}$ — множество булевых функций ($f_i \neq f_j$ при $i \neq j$).

Определение. Многополюсник из Ф. Э., имеющий n входов и s выходов, называется *универсальным* для данного множества функций, если для каждого $i (1 \leqslant i \leqslant s)$ в многополюснике найдется выход $\tau(i)$ такой, что на нем реализуется функция $f_i(x_1, \ldots, x_n)$.

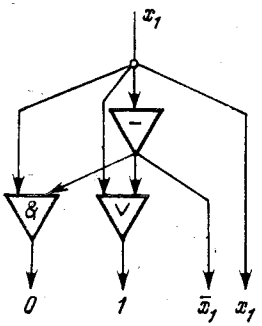


Рис. 24

Пример 2. Пусть $\{K_1, \ldots, K_s\}$ — множество конъюнкций. Тогда многополюсник (см. рис. 20) является универсальным для этого множества.

Лемма 4. *Для любого n можно построить универсальный многополюсник U_n для множества всех булевых функций от n переменных $x_1, \ldots, x_n$, и*

$$L(U_n) \leqslant 2 \cdot 2^{2^n}.$$

Доказательство. Построение многополюсника U_n будем осуществлять индуктивным способом.

Базис индукции ($n = 1$). В качестве U_1 возьмем многополюсник, изображенный на рис. 24. Мы имеем $L(U_1) = 3 < 2 \cdot 2^{2^1}$.

Индуктивный переход. Предположим, что построен универсальный многополюсник U_{n-1} для множества всех булевых функций, зависящих от переменных $x_1, \ldots, x_{n-1}$, и $L(U_{n-1}) \leqslant 2 \cdot 2^{2^{n-1}}$. Рассмотрим разложение

$$f(x_1, \ldots, x_{n-1}, x_n) = x_n f'(x_1, \ldots, x_{n-1}) \vee \bar{x}_n f''(x_1, \ldots, x_{n-1}).$$

Множество всех булевых функций, зависящих от переменных $x_1, \ldots, x_n$, разобьем на три непересекающихся класса.

I. $f' \equiv f'' \equiv 0$. Этот класс содержит одну функцию $f \equiv 0$.

II. Ровно одна из функций f' или f'' тождественно равна 0. В этом классе содержатся функции f вида

$$f = x_n f'(x_1, \ldots, x_{n-1}) \quad (f' \not\equiv 0)$$

или

$$f = \bar{x}_n f''(x_1, \ldots, x_{n-1}) \quad (f'' \not\equiv 0),$$

т. е. $2(2^{2^{n-1}} - 1)$ функций.

III. Все остальные функции, т. е. функции f, у которых

$$f' \neq 0 \text{ и } f'' \neq 0.$$

В этом классе имеется

$$2^{2^n} - 2(2^{2^{n-1}} - 1) - 1$$

функций.

На рис. 25 изображен многополюсник U_n. Здесь выходы многополюсника U_{n-1} занумерованы числами

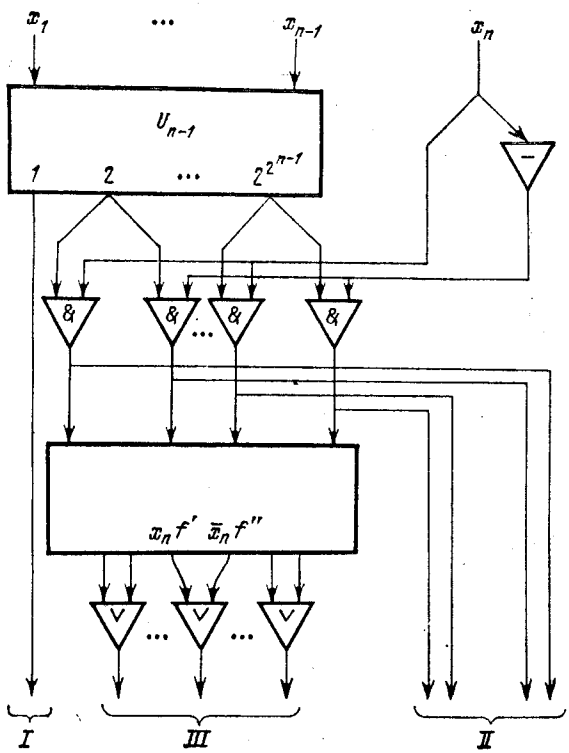


Рис. 25

$1, \ldots, 2^{2^{n-1}}$, причем считается, что на выходе 1 реализуется константа 0.

Выходы многополюсника U_n разбиты на три класса в соответствии с разбиением множества всех булевых функций, зависящих от переменных $x_1, \ldots, x_n$. Данный многополюсник содержит:

а) подсхему U_{n-1} и вне ее еще
б) один инвертор,
в) $2\left(2^{2^{n-1}}-1\right)$ конъюнкторов,
г) $2^{2^n}-2\left(2^{2^{n-1}}-1\right)-1$ дизъюнкторов.
Таким образом,

$$L(U_n)=L(U_{n-1})+1+2\left(2^{2^{n-1}}-1\right)+2^{2^n}-2\left(2^{2^{n-1}}-1\right)-$$
$$-1=L(U_{n-1})+2^{2^n}\leqslant 2\cdot 2^{2^{n-1}}+2^{2^n}\leqslant 2\cdot 2^{2^n}.$$

Лемма доказана.

Теорема 6 (К. Э. Шеннон [34]). *Существует метод синтеза (алгоритм A_4), который для каждой булевой функции $f(x_1, \ldots, x_n)$ позволяет построить схему из Ф. Э. сложности $L_{A_4}(f)$ и*

$$L_{A_4}(f(x_1, \ldots, x_n))\leqslant 8\frac{2^n}{n}.$$

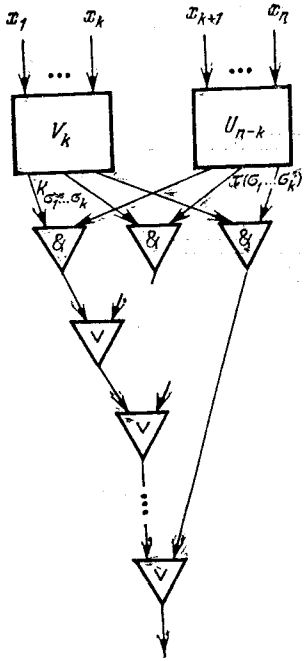


Рис. 26

Доказательство. Возьмем разложение

$$f(x_1, \ldots, x_n)=\bigvee_{(\sigma_1,\ldots,\sigma_k)} x_1^{\sigma_1}\& \ldots \& x_k^{\sigma_k}\ \&$$
$$\&\ f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Рассмотрим схему Σ_f, изображенную на рис. 26.

V_k — многополюсник, универсальный для множества всех конъюнкций $K_{\sigma_1\ldots\sigma_k}$, где

$$K_{\sigma_1\ldots\sigma_k}=x_1^{\sigma_1}\& \ldots \& x_k^{\sigma_k}$$

(k — фиксированное число). В качестве V_k взята схема Σ^k (см. рис. 22).

U_{n-k} — многополюсник, универсальный для множества всех булевых функций, зависящих от переменных $x_{k+1}, \ldots, x_n$. Через $\tau(\sigma_1, \ldots, \sigma_k)$ обозначен его выход, на котором реализуется функция

$$f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Легко видеть, что данная схема Σ_f реализует функцию $f(x_1, \ldots, x_n)$ в соответствии с приведенным выше разложением. Оценим сложность Σ_f:

$$L(\Sigma_f) \leqslant L(V_k) + L(U_{n-k}) + 2 \cdot 2^k - 1 \leqslant 2 \cdot 2^k + k - \\ - 4 + 2 \cdot 2^{2^{n-k}} + 2 \cdot 2^k - 1 = 4 \cdot 2^k + 2 \cdot 2^{2^{n-k}} + k - 5.$$

Подберем параметр k таким образом, чтобы правая часть этого неравенства стала возможно меньше. Так как нас интересует порядок величины $L_{A_4}(f)$, то, вместо минимума правой части, который в дискретном случае вычисляется сложно, мы возьмем значение правой части при $m = [\log_2(n - 2\log_2 n)]$, где $m = n - k$. Это значение подбирается, исходя из следующих соображений: с ростом k член $4 \cdot 2^k$ возрастает, а член $2 \cdot 2^{2^{n-k}}$ убывает, и минимум достигается, когда оба члена становятся близкими друг к другу.

Мы имеем $\frac{1}{2}(n - 2\log n) < 2^m < (n - 2\log n)$,

$$L_{A_4}(f) \leqslant 4\frac{2^n}{2^m} + 2 \cdot 2^{2^m} \leqslant \frac{4 \cdot 2^n}{\frac{1}{2}(n - 2\log n)} + 2\frac{2^n}{n^2} \leqslant 8\frac{2^n}{n}.$$

Следствие. *Учитывая нижнюю оценку, мы получаем*

$$L(n) \asymp \frac{2^n}{n},$$

и, значит, алгоритм A_4 является наилучшим по порядку.

§ 6. Асимптотически наилучший метод синтеза схем из Ф. Э. (метод Лупанова)

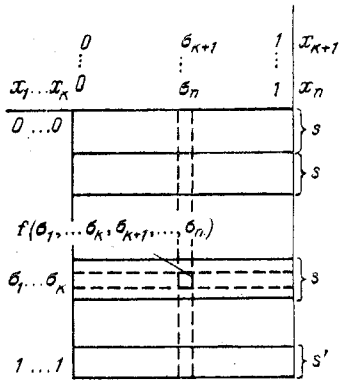
Первым асимптотически наилучшим методом синтеза для реализации всех булевых функций был метод Лупанова, который был сначала сформулирован для контактных схем и затем им же был распространен на схемы из Ф. Э.

Теорема 7 (О. Б. Лупанов [18]). *Для схем из Ф. Э. в базисе, состоящем из инвертора, дизъюнктора и конъюнктора, можно построить асимптотически наилучший метод синтеза и*

$$L(n) \sim \frac{2^n}{n}.$$

Доказательство. Зададим произвольную булеву функцию $f(x_1, \ldots, x_n)$ при помощи таблицы размера $2^k \times 2^{n-k}$ (см. табл. 3).

Таблица 3



Строки этой таблицы нумеруем наборами значений по переменным $x_1, \ldots, x_k$, столбцы — по переменным $x_{k+1}, \ldots$ $\ldots, x_n$. На пересечении строки $(\sigma_1, \ldots, \sigma_k)$ и столбца $(\sigma_{k+1} \ldots \sigma_n)$ помещаем значение $f(\sigma_1, \ldots, \sigma_k, \sigma_{k+1}, \ldots, \sigma_n)$.

Легко видеть, что столбец с номером $\sigma_{k+1}, \ldots, \sigma_n$ задает функцию $f(x_1, \ldots, x_k, \sigma_{k+1}, \ldots, \sigma_n)$, являющуюся компонентой разложения

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_{k+1}, \ldots, \sigma_n)} x_{k+1}^{\sigma_{k+1}} \ldots x_n^{\sigma_n} f(x_1, \ldots, x_k, \sigma_{k+1}, \ldots, \sigma_n). \qquad (1)$$

Возьмем целое число s, $1 < s < 2^k$, и разрежем таблицу на полосы шириной s (см. табл. 3). При этом последняя полоса может оказаться меньшей ширины s' $(s' \leqslant s)$. Занумеруем сверху вниз полученные полосы числами 1, 2, ..., p, где $p =]2^k/s[$, и рассмотрим полосу с номером i (см. табл. 4) и строками $(\sigma_1(1) \ldots \sigma_k(1)), \ldots$ $\ldots, (\sigma_1(s) \ldots \sigma_k(s))$.

Эта полоса распадается на короткие столбцы высоты s для последней полосы — высоты s'. Поэтому число $t(i)$ сортов коротких столбцов будет не более 2^s. Произведем нумерацию этих сортов числами 1, 2, ..., $t(i)$.

Пусть

$(\gamma_1, \ldots, \gamma_s)$ — столбец j-го сорта.

Обозначим через $f_{ij}(x_1, \ldots, x_k)$ булеву функцию, определя-

емую этим коротким столбцом:

$$f_{ij}(x_1, \ldots, x_k) = \begin{cases} \gamma_l, \text{ если } (\sigma_1 \ldots \sigma_k) = (\sigma_1(l) \ldots \sigma_k(l)), \\ \qquad l = 1, \ldots, s, \\ 0, \text{ если } (\sigma_1 \ldots \sigma_k) \text{ не принадлежит} \\ \qquad i\text{-й полосе.} \end{cases}$$

Столбец с номером $\sigma_{k+1} \ldots \sigma_n$ разрезается полосами на p коротких столбцов. Поэтому

$$f(x_1, \ldots, x_k, \sigma_{k+1}, \ldots, \sigma_n) = \\ = f_{1j_1}(x_1, \ldots, x_k) \vee \ldots \vee f_{pj_p}(x_1, \ldots, x_k), \quad (2)$$

где j_i — номер сорта соответствующего короткого столбца, принадлежащего i-й полосе.

Таблица 4

		j	
$\sigma_1(1) \ldots \sigma_k(1)$		γ_1	
		.	
		.	
		.	
$\sigma_1(s) \ldots \sigma_k(s)$		γ_s	

Теперь перейдем к описанию схемы Σ. Ее мы получим в виде соединения отдельных блоков (см. рис. 27). Попутно будем оценивать сложность блоков.

1. Блок A реализует все конъюнкции $x_1^{\sigma_1} \ldots x_k^{\sigma_k}$

$$L(A) \leqslant k2^k.$$

2. Блок B реализует все конъюнкции $x_{k+1}^{\sigma_{k+1}} \ldots x_n^{\sigma_n}$

$$L(B) \leqslant (n-k)2^{n-k}.$$

3. Блок C реализует по совершенной д. н. ф. функции $f_{ij}(x_1, \ldots, x_k)$

$$L(C) \leqslant (s-1)(t(1) + \ldots \\ \ldots + t(p)) < sp\,2^s.$$

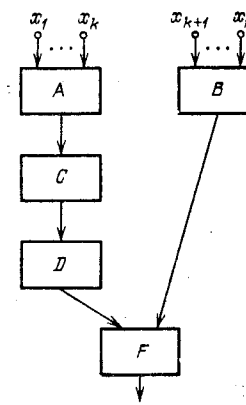


Рис. 27

4. Блок D реализует функции $f(x_1, \ldots, x_k, \sigma_{k+1}, \ldots, \sigma_n)$ на основе формулы (2)

$$L(D) \leqslant (p-1)2^{n-k} < p2^{n-k}.$$

5. Блок F реализует функцию $f(x_1, \ldots, x_n)$ исходя из ее разложения (1)

$$L(F) \leqslant 2^{n-k} + 2^{n-k} - 1 < 2 \cdot 2^{n-k}.$$

Суммируя полученные оценки, имеем

$$L(\Sigma) \leqslant k2^k + (n-k)2^{n-k} + sp2^s + p2^{n-k} + 2 \cdot 2^{n-k}.$$

Положим $k = [3 \log_2 n]$, $s = [n - 5 \log_2 n]$. Тогда

$$p = \left]\frac{2^k}{s}\right[\sim \frac{2^k}{s}$$

и

$$L(\Sigma) \leqslant k2^k + n2^{n-k} + 2^{k+s} + \frac{2^n}{s},$$

причем

$$k2^k = o\left(\frac{2^n}{n}\right),$$

$$n2^{n-k} = o\left(\frac{2^n}{n}\right),$$

$$2^{k+s} = o\left(\frac{2^n}{n}\right),$$

$$\frac{2^n}{s} \sim \frac{2^n}{n}.$$

Поэтому

$$L(\Sigma) \leqslant \frac{2^n}{n}.$$

В силу произвольности функции $f(x_1, \ldots, x_n)$ отсюда следует, что

$$L(n) \leqslant \frac{2^n}{n}.$$

Соединяя полученные соотношения с теоремой 2, мы завершаем доказательство.

§ 7. Синтез сумматора

Общая теория синтеза схем из Ф. Э. приводит к важному выводу о том, что большинство булевых функций (при $n \to \infty$) имеет сложные минимальные схемы. Это означает, что практическую ценность с точки зрения синтеза представляет весьма узкий класс булевых функций. Поэтому наряду с универсальными методами синтеза необходимо иметь методы синтеза, приспособлен-

ные к отдельным классам булевых функций, полнее учитывающие свойства отдельных функций.

В этом параграфе мы познакомимся с одним из подходов к реализации достаточно узкого класса функций. Речь пойдет о построении многополюсной схемы из функциональных элементов, реализующей сложение двух чисел, заданных в двоичной системе счисления

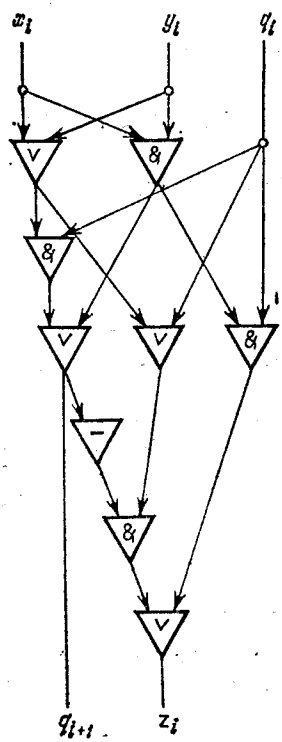


Рис. 28

$$x = x_n x_{n-1} \ldots x_1,$$

$$y = y_n y_{n-1} \ldots y_1.$$

Для этого рассмотрим хорошо известный алгоритм сложения чисел x и y «столбиком»

$$\begin{array}{r} (q_{n+1}\, q_n \ldots q_1) \\ x_n \ldots x_1 \\ + \quad y_n \ldots y_1 \\ \hline z_{n+1}\, z_n \ldots z_1 \end{array}.$$

Здесь числа $q_{n+1}, \ldots, q_1$ обозна-

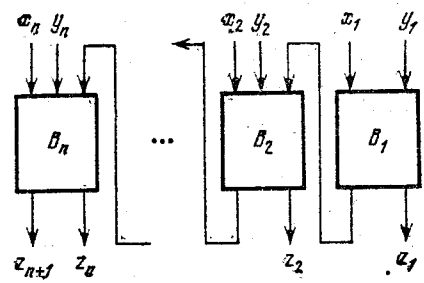


Рис. 29

чают результаты переносов из предыдущих разрядов ($q_1 = 0$). Очевидно,

$$z_i = x_i + y_i + q_i (\mathrm{mod}\ 2),$$

$$q_{i+1} = x_i y_i \vee x_i q_i \vee y_i q_i.$$

Основываясь на тождестве

$$x_i + y_i + q_i = \overline{(x_i y_i \vee x_i q_i \vee y_i q_i)}\,\&\,(x_i \vee y_i \vee q_i) \vee x_i y_i q_i,$$

легко получить схему, реализующую соответствующее

преобразование величин x_i, y_i, q_i в z_i, q_{i+1} (см. рис. 28). Обозначим данную схему через $B_i (1 < i \leqslant n)$. Тогда искомая схема Σ_n — сумматор для двух n-разрядных двоичных чисел — получается путем последовательного соединения блоков B_i (см. рис. 29). Здесь $z_{n+1} = q_{n+1}$, и блок B_1 осуществляет преобразование

$$z_1 = x_1 + y_1 = \overline{x_1 y_1}(x_1 \vee y_1),$$

$$q_2 = x_1 y_1.$$

Очевидно, $L(B_1) = 4$ и $L(B_i) = 9$ при $1 < i \leqslant n$. Таким образом,

$$L(\Sigma_n) \leqslant 9(n-1) + 4 = 9n - 5 < 9n.$$

§ 8. Синтез схем из Ф. Э., реализующих симметрические функции

Другим интересным классом двоичных преобразований является класс так называемых симметрических функций.

Определение. Булева функция $S(x_1, \ldots, x_n)$ называется *симметрической*, если она является симметрической относительно всех переменных $x_1, \ldots, x_n$, т. е. если для любого набора $(\alpha_1, \ldots, \alpha_n)$ и любой подстановки $\begin{pmatrix} 1 & \ldots & n \\ j_1 & \ldots & j_n \end{pmatrix}$ имеет место

$$S(\alpha_{j_1}, \ldots, \alpha_{j_n}) = S(\alpha_1, \ldots, \alpha_n).$$

Классу симметрических функций, очевидно, принадлежат константы 0 и 1, функции $\bar{x}_1$, $x_1 \& x_2$, $x_1 \vee x_2$, $x_1 + x_2$, $x_1 + x_2$, $x_1 x_2 \vee x_1 x_3 \vee x_2 x_3$ и т. д.

Легко видеть, что если для набора $(\alpha_1, \ldots, \alpha_n)$

$$S(\alpha_1, \ldots, \alpha_n) = 1,$$

и $(\alpha_1, \ldots, \alpha_n)$ содержит ровно i единиц, то для любого набора $(\alpha'_1, \ldots, \alpha'_n)$, содержащего также ровно i единиц,

$$S(\alpha'_1, \ldots, \alpha'_n) = 1.$$

В таком случае симметрическая функция $S(x_1, \ldots, x_n)$ характеризуется списком своих рабочих чисел $i_1, \ldots, i_r (0 \leqslant i_1 < \ldots < i_r \leqslant n)$, обозначающих число единиц в наборах $(\alpha_1, \ldots, \alpha_n)$, для которых

$$S(\alpha_1, \ldots, \alpha_n) = 1.$$

Поэтому можно писать

$$S(x_1, \ldots, x_n) = S_{i_1,\ldots,i_r}(x_1, \ldots, x_n),$$

где индексы $i_1, \ldots, i_r$ — рабочие числа функции $S(x_1, \ldots, x_n)$. Например,

$$x_1x_2 \vee x_1x_3 \vee x_2x_3 = S_{2,3}(x_1, x_2, x_3).$$

Рассмотрим вспомогательное преобразование

$$(\alpha_1, \ldots, \alpha_n) \to i_{(\alpha_1,\ldots,\alpha_n)},$$

где $i_{(\alpha_1,\ldots,\alpha_n)}$ — число единиц в наборе $(\alpha_1, \ldots, \alpha_n)$. Это преобразование будет реализовано многополюсником Σ'_n из функциональных элементов (см. рис. 30), в котором на выходах $z_1, \ldots, z_{[\log_2 n]+1}$ появляется двоичная запись числа $i_{(\alpha_1,\ldots,\alpha_n)}$, когда на вход поступает набор $(\alpha_1, \ldots, \alpha_n)$.

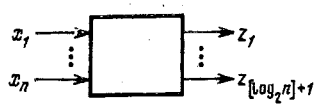


Рис. 30

Пусть $n = 2^m$. Тогда $[\log_2 n] + 1 = m + 1$.

Лемма 5. *Можно построить многополюсник* Σ'_n, *осуществляющий в вышеуказанном смысле преобразование*

$$(\alpha_1, \ldots, \alpha_n) \to i_{(\alpha_1,\ldots,\alpha_n)},$$

и $L(\Sigma'_n) \leqslant 18n - 9\log n - 18$.

Доказательство проводится индукцией по m.

Рис. 31

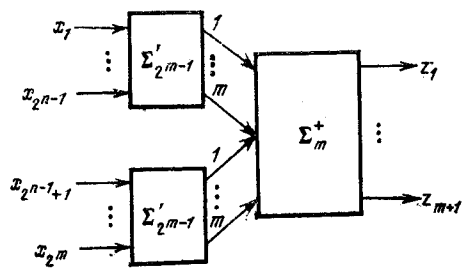


Рис. 32

а) Базис индукции $m = 0$. Здесь $n = 1$ и искомая схема Σ'_1 имеет вид (см. рис. 31), $L(\Sigma'_1) = 0$. С другой стороны, $18n - 9\log_2 n - 18 = 0$ при $n = 1$.

Следовательно, неравенство

$$L(\Sigma'_n) \leqslant 18n - 9\log n - 18$$

справедливо для $n = 1$.

б) Индуктивный переход от $m-1$ к $m\,(m-1 \geqslant 0)$ осуществляется при помощи схемы Σ'_{2^m} (см.

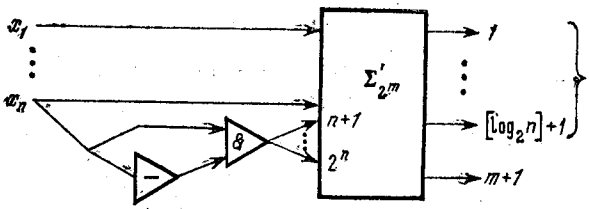


Рис. 33

рис. 32). Эта схема «делит» набор $(\alpha_1, \ldots, \alpha_{2^m})$ на две части $(\alpha_1, \ldots, \alpha_{2^{m-1}})$ и $(\alpha_{2^{m-1}+1}, \ldots, \alpha_{2^m})$, в каждой части под схемами $\Sigma'_{2^{m-1}}$ вычисляется число единиц i' и i'', затем сумматор Σ^+_m «складывает» i' и i'', и на выходе вычисляется число единиц исходного набора.

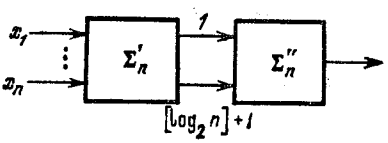


Рис. 34

Мы имеем $L(\Sigma'_{2^m}) = 2L(\Sigma'_{2^{m-1}}) + L(\Sigma^+_m)$, или

$$L(\Sigma'_n) = 2L(\Sigma'_{n/2}) + L(\Sigma^+_m) \leqslant 2\left(18\frac{n}{2} - 9\log\frac{n}{2} - 18\right) + 9\log_2 n < 18n - 9\log n - 18.$$

Лемма доказана.

Следствие. $L(\Sigma'_{2^m}) \leqslant 18 \cdot 2^m$, *т. е. при* $n = 2^m$

$$L(\Sigma'_n) \leqslant 18n.$$

Лемма 6. *Для любого* n *можно построить многополюсник* Σ'_n, *осуществляющий преобразование* $(\alpha_1, \ldots, \alpha_n) \to i_{(\alpha_1, \ldots, \alpha_n)}$, *и* $L(\Sigma'_n) \leqslant 36n$.

Доказательство. Очевидно, существует такое m, что

$$n \leqslant 2^m < 2n.$$

Искомый многополюсник Σ'_n может быть реализован так, как это показано на рис. 33. У многополюсника Σ'_{2^m} берутся первые $[\log_2 n] + 1$ выходов:

$$L(\Sigma'_n) = L(\Sigma'_{2^m}) + 2 \leqslant 18 \cdot 2^m + 2 \leqslant 18(2n-1) + 2 \leqslant 36n.$$

Утверждение доказано.

Теорема 8. *Существует константа C_0 такая, что любая симметрическая функция $S_{i_1,\ldots,i_r}(x_1, \ldots, x_n)$ может быть реализована схемой $\Sigma_{S(x_1,\ldots,x_n)}$ такой, что* $L(\Sigma_{S(x_1, \ldots, x_n)}) \leqslant C_0 n$.

Доказательство. Искомая схема $\Sigma_{(x_1,\ldots,x_n)}$ может быть построена так, как это показано на рис. 34. Здесь схема Σ'_n преобразует набор $(\alpha_1, \ldots, \alpha_n)$ в двоичный код $i_{(\alpha_1,\ldots,\alpha_n)}$. Схема Σ''_n реализует булеву функцию от $[\log_2 n] + 1$ переменной, и эта функция на «наборах» $i_{(\alpha_1,\ldots,\alpha_n)}$, совпадающих с $i_1, \ldots, i_r$, равна 1, а в остальных случаях равна нулю.

Мы имеем (см. теорему 7)

$$L(\Sigma_{S(x_1, \ldots, x_n)}) \leqslant 36n + \frac{2^{[\log_2 n]+1}}{[\log_2 n]+1} \leqslant 36n + \frac{16n}{\log_2 n},$$

откуда следует существование константы C_0, о которой говорилось в теореме. Теорема доказана.

СПИСОК ЛИТЕРАТУРЫ

1. Васильев Ф. П. Численные методы решения экстремальных задач.— М.: Наука, 1980.
2. Васильев Ю. Л. О сравнении сложности тупиковых и минимальных дизъюнктивных нормальных форм.— В кн.: Проблемы кибернетики. Вып. 10.— М.: Физматгиз, 1963, с. 5—61.
3. Гаврилов Г. П., Сапоженко А. А. Сборник задач по дискретной математике.— М.: Наука, 1977.
4. Дискретная математика и математические вопросы кибернетики/Под ред. С. В. Яблонского и О. Б. Лупанова.— М.: Наука, 1974.
5. Журавлев Ю. И. Об отделимости подмножеств вершин n-мерного единичного куба.— В кн.: Труды МИАН СССР. Т. 51.— М.: Изд-во АН СССР, 1958, с. 143—157.
6. Журавлев Ю. И. Об алгоритмах упрощения дизъюнктивных нормальных форм.— ДАН СССР, 1960, 132, № 2, с. 260—263.
7. Зыков А. А. Гиперграфы.— УМН, 1974, 29, № 6, с. 89—154.
8. Ильин В. А., Садовничий В. А., Сендов Б. Х. Математический анализ.— М.: Наука, 1979.
9. Кратко М. И. Алгоритмическая неразрешимость одной задачи из теории конечных автоматов.— В кн.: Дискретный анализ. Вып. 2.— Новосибирск, 1964, с. 37—41.
10. Кратко М. И. О существовании нерекурсивных базисов конечных автоматов.— Алгебра и логика, 1964, 3, № 2, с. 33—44.
11. Кудрявцев В. Б. Теорема полноты для одного класса автоматов без обратных связей.— В кн.: Проблемы кибернетики. Вып. 8.— М.: Физматгиз, 1962, с. 91—115.
12. Кудрявцев В. Б. О мощностях множеств предполных множеств некоторых функциональных систем, связанных с автоматами.— В кн.: Проблемы кибернетики. Вып. 13. М.: Наука, 1965, с. 45—74.
13. Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов.— М.: Наука, 1986.
14. Кудрявцев Л. Д. Курс математического анализа. Т. 1—2.— М.: Высш. шк., 1981.
15. Кузнецов А. В. О проблемах тождества и функциональной полноты алгебраических систем.— В кн.: Труды 3-го Всесоюзного мат. съезда. Т. 2.— М.: Изд-во АН СССР, 1956, с. 145—146.
16. Кузнецов А. В. О бесповторных контактных схемах и бесповторных суперпозициях функций алгебры логики.— В кн.: Труды МИАН СССР. Т. 51.— М.: Изд-во АН СССР, 1958, с. 186—225.

17. Лупанов О. Б. О возможностях синтеза схем из произвольных элементов.— В кн.: Труды МИАН СССР. Т. 51.— М.: Изд-во АН СССР, 1958, с. 158—173.
18. Лупанов О. Б. О синтезе некоторых классов управляющих систем.— В кн.: Проблемы кибернетики. Вып. 10.— М.: Физматгиз, 1963, с. 63—97.
19. Лупанов О. Б. Об одном подходе к синтезу управляющих систем — принципе локального кодирования.— В кн.: Проблемы кибернетики. Вып. 14.— М.: Наука, 1965, с. 31—110.
20. Лупанов О. Б. Асимптотические оценки сложности управляющих систем.— М.: Изд-во МГУ, 1984.
21. Ляпунов А. А. О логических схемах программ.— В кн.: Проблемы кибернетики. Вып. 1.— М.: Физматгиз, 1958, с. 46—74.
22. Марков Ал. А. Об алфавитном кодировании. I.— ДАН СССР, 1960. 132, № 3, с. 521—523.
23. Марков Ал. А. Об алфавитном кодировании. II.— ДАН СССР, 1961. 139, № 3, с. 560—561.
24. Марков Ал. А. Нерекуррентное кодирование.— В кн.: Проблемы кибернетики. Вып. 8.— М.: Физматгиз, 1962, с. 169—186.
25. Марков А. А. Введение в теорию кодирования.— М.: Наука, 1982.
26. Никольский С. М. Курс математического анализа. Т. 1—2.— М.: Наука, 1983.
27. Орлов В. А. Простое доказательство алгоритмической неразрешимости некоторых задач о полноте автоматных базисов.— Кибернетика, 1973, № 4, с. 109—113.
28. Понтрягин Л. С. Основы комбинаторной топологии.— М.: Наука, 1976.
29. Рыбников К. А. Введение в комбинаторный анализ.— М.: Изд-во МГУ, 1985.
30. Сачков В. Н. Введение в комбинаторные методы дискретной математики.— М.: Наука, 1982.
31. Трахтенброт Б. А. К теории бесповторных контактных схем.— В кн.: Труды МИАН СССР. Т. 51.— М.: Изд-во АН СССР, 1958, с. 226—269.
32. Холл М. Комбинаторика.— М.: Мир, 1970.
33. Чегис И. А., Яблонский С. В. Логические способы контроля работы электрических схем.— В кн.: Труды МИАН СССР. Т. 51.— М.: Изд-во АН СССР, 1958, с. 270—360.
34. Шеннон К. Работы по теории информации и кибернетике.— М.: ИЛ, 1963.
35. Яблонский С. В. О суперпозициях функций алгебры логики.— Матем. сб., 1952, 30 (72), № 2, с. 329—348.
36. Яблонский С. В. Функциональные построения в k-значной логике.— В кн.: Труды МИАН СССР. Т. 51.— М.: Изд-во АН СССР, 1958, с. 5—142.
37. Яблонский С. В. Методические разработки по курсу «Элементы дискретной математики».— М.: Изд-во МГУ, 1971.
38. Яблонский С. В., Гаврилов Г. П., Кудрявцев В. Б. Функции алгебры логики и классы Поста.— М.: Наука, 1966.
39. Янов Ю. И., Мучник А. А. О существовании k-значных замкнутых классов, не имеющих конечного базиса.— ДАН СССР, 1959, 127, № 1, с. 44—46.
40. Hamming R. W. Error-detecting and error-correcting codes.— Bell Syst. Techn. J., 1950, 29, № 2, p. 147—160.

[Рус. пер.: Коды с обнаружением и исправлением ошибок.— М.: ИЛ, 1956, с. 7—22.]
41. Huffman D. A. A method for the construction of minimum-redundancy codes.— Proc. IRE, 1952.— 40, № 9, p. 1098—1101 [Рус. пер.: Кибернетический сборник. Вып. 3.— М.: ИЛ, 1961, с. 79—87.]
42. Quine W. V. On cores and prime implicants of truth functions.— Amer. Math. Monthly, 1959.— 66, № 9, p. 755—760.
43. McMillan B. Two inequalities implied by unique decipherability.— IRE Trans., 1956.— IT—2, № 4, p. 115—116 [Рус. пер.: Кибернетический сборник. Вып. 3.— М.: ИЛ, 1961, с. 88—92.]
44. Martin N. M. The Sheffer functions of 3-valued logic.— J. Symb. Logic., 1954, 19, № 1, p. 45—51.
45. Piccard S. Sur les fonctions définies dans les ensembles finis quelconque.— Fund. Math., 1955, 24, p. 183—185.
46. Post E. L. Introduction to a general theory of elementary propositions.— Amer. J. Math., 1921, 43, № 3, p. 163—185.
47. Post E. L. The two-valued iterative systems of mathematical logic.— Annals of Math. Studies, v. 5, Princeton Univ. Press, Princeton — London, 1941.
48. Rosser J. B., Turquette A. R. Many-valued logics.— Amsterdam, 1952.
49. Salomaa A. Some completeness criteria for sets of functions over a finite domain I, II.— Turun Ylopiston Jalkaisuja Annales Universitatis Turkuensis, 1962, sarja A53, p. 1—9; 1963, sarja A63, p. 1—19. [Рус. пер.: Кибернетический сборник. Вып. 8.— М.: Мир, 1964, с. 7—32.]
50. Slupecki J. Kriterium pelnosci wielowar — tosciowych systemow logiki zdan.— Comptes Rendus des Séances de la Société des Sciences et des Lettres de Varsovie, Cl. III, 1939, 32, p. 102—128.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

УКАЗАТЕЛЬ ОБОЗНАЧЕНИЙ