Министерство общего и профессионального образования РФ
Московский физико-технический институт
(государственный университет)

Ю.И. Журавлёв, Ю.А. Флёров

ДИСКРЕТНЫЙ АНАЛИЗ
Ч.1

Рекомендовано Учебно-методическим советом
Московского физико-технического института
(государственного университета)
в качестве учебного пособия для студентов высших учебных заведений
по специальности "Прикладная математика и физика"

Москва 1999

Ж 91
УДК 519.1

Рецензенты:
кафедра математической кибернетики ВМК
МГУ им. Ломоносова

чл.—корр. РАН К.В. Рудаков

Журавлёв Ю.И., Флёров Ю.А.
Ж 91 Дискретный анализ. Ч. 1: учебное пособие/МФТИ.М., 1999. 136 с.

ISBN 5-7417-0108-6

Содержит материал, излагаемый в первом семестре курса дискретного анализа: комбинаторике, элементы алгебры логики, начальные сведения теории графов. Включены как основополагающие разделы, предназначенные для первоначального ознакомления с теорией, так и материал повышенной трудности, непосредственно в лекциях не излагаемый. Рассмотрены оценки сложности решений многих задач.

Предназначено для студентов I курса факультета прикладной математики и экономики Московского физико-технического института. Может быть использовано студентами для изучения соответствующих разделов программы, преподавателями для подготовки упражнений, заданий, семинарских занятий, экзаменационного материала, а также для самостоятельного ознакомления и изучения основных разделов дискретного анализа

Библиогр.: 14 назв.

Учебное издание
Журавлёв Юрий Иванович
Флёров Юрий Арсениевич
ДИСКРЕТНЫЙ АНАЛИЗ
Ч.1
Редактор **И. А. Волкова**
Лицензия ЛР № 040060 от 21.08.96
Подписано в печать 07.05.99. Формат 60×901/16.
Печать офсетная. Усл. п.л. 8,5. Уч.-изд.л. 8,75. Тираж 400 экз. Заказ N
Московский физико-технический институт
(государственный университет)
Лаборатория обработки учебной и научной информации
141700, Моск. обл., г. Долгопрудный, Институтский пер., 9

ISBN 5-7417-0108-6

Оглавление

Глава 1

Элементы комбинаторики

1.1 Введение

Сейчас трудно было бы, пожалуй, назвать раздел теоретической информатики, в котором в течение последнего десятилетия были бы достигнуты большие успехи, нежели в конструировании и анализе комбинаторных алгоритмов — важнейшей проблемы комбинаторики. С одной стороны, было обнаружено много новых, более эффективных методов решения комбинаторных задач с помощью ЭВМ, с другой стороны — получены теоретические результаты, свидетельствующие все более явно о том, что для широкого класса проблем не существует достаточно эффективных алгоритмов. Эффективные комбинаторные алгоритмы находят применение во многих областях нечисленной обработки информации, особенно в дискретной оптимизации и в исследовании операций, в вычислительной математике. Количество комбинаторных задач и их разнообразие быстро растет. К их решению прямо или косвенно приводят многие практические задачи. Во многих областях математики (теория графов, теория чисел, теория групп, кибернетика, вычислительная математика) имеются задачи или группы задач, комбинаторный характер которых угадывается без усилий. Между этими задачами часто можно установить сеть взаимных интерпретаций, что приводит к мысли о наличии их общей теоретической основы. При этом оказывается, что несмотря на заманчивую простоту постановки, комбинаторные задачи в большинстве очень трудны. Комбинаторика — раздел математики, в котором изучаются вопросы о том, сколько различных конфигураций (комбинаций), подчиненных тем или иным условиям, мож-

но составить из заданных объектов. Основная проблема комбинаторики — подсчет числа элементов в конечном множестве. В этой широкой проблеме можно условно выделить следующие основные направления исследований:

- изучение известных конфигураций;
- исследование неизвестных конфигураций;
- подсчет числа конфигураций;
- приближенный подсчет числа конфигураций;
- перечисление конфигураций;
- оптимизация.

1.2 Два принципа комбинаторики

При подсчете числа различных комбинаций в комбинаторике используются следующие два основных правила.

Правило произведения: если объект A может быть выбран m различными способами и после каждого из таких выборов объект B в свою очередь может быть выбран n различными способами, то выбор двух объектов «A и B» в указанном порядке может быть осуществлен mn способами.

Правило суммы: если объект A может быть выбран m различными способами, а объект B может быть выбран другими n различными способами при условии, что одновременный выбор A и B невозможен, то выбор «A или B» может быть осуществлен $m + n$ способами.

1.3 Функции и размещения

Классической задачей комбинаторики является задача определения числа способов размещения некоторых объектов в каком-то количестве ящиков так, чтобы были выполнены заданные ограничения. Эту задачу можно сформулировать несколько более формально следующим образом.

Термины «функция», «отображение», «преобразование» и «соответствие» будут в дальнейшем использоваться как синонимы.
При этом запись $f : X \to Y$ означает, что f есть функция с областью определения X, область значений которой содержится во множестве Y,

то есть для каждого $x \in X$ функция f определяет единственный элемент $y = f(x) \in Y$.

Пусть даны множества X и Y, причем множество X содержит n элементов ($|X| = n$), а множество Y содержит m элементов ($|Y| = m$). В этих терминах задача может быть сформулирована следующим образом: сколько существует функций (отображений), удовлетворяющих заданным ограничениям. Элементы множества X соответствуют объектам, элементы множества Y — ящикам, а каждая функция $f : X \to Y$ определяет некоторое размещение, указывая для каждого объекта $x \in X$ ящик $y = f(x) \in Y$, в котором данный объект находится.

Другую традиционную интерпретацию можно получить, трактуя Y как множество цветов, а $f(x)$ как «цвет объекта x». Наша задача эквивалентна, таким образом, вопросу, сколькими способами можно покрасить объекты так, чтобы были соблюдены некоторые ограничения.

Наконец, каждому отображению $f : X \to Y$, $|X| = n$, $|Y| = m$, можно взаимно однозначно сопоставить слово $< f(x_1), \ldots, f(x_n) >= =< y_1, y_2, \ldots, y_n >= y_1 y_2 \ldots y_n$ в алфавите из m символов. Получаем третью эквивалентную формулировку задачи: подсчет числа слов в алфавите, удовлетворяющих заданным ограничениям.

Самой простой является задача, в которой на рассматриваемые функции не накладывается никаких ограничений.

Утверждение 1.1. *Если $|X| = n$, $|Y| = m$, то количество всех функций $f : X \to Y$ равно mn.*

Эквивалентное утверждение 1.1': число слов длины n в алфавите из m символов равно mn.

Доказательство. Без потери общности можно всегда считать, что $X = 1, \ldots, n$, $Y = 1, \ldots, m$. Каждую функцию можно тогда отождествить с последовательностью $< f(1), \ldots, f(n) >=< y_1, \ldots, y_n >$. Каждый член y_i последовательности можно выбрать m способами, что дает mn возможностей выбора последовательности $< y_1, \ldots, y_n >$. □

Определение. Отображение $f : X \to Y$ сюръективно, если для каждого элемента $y \in Y$ существует хотя бы один элемент $x \in X$, такой что $\varphi(x) = y$ (в каждом ящике при размещении находится хотя бы один объект, все буквы алфавита используются в слове, все цвета используются при окраске).

Определение. Отображение $\varphi : X \to Y$ инъективно, если $x_1 \neq x_2 \Rightarrow \varphi(x_1) \neq \varphi(x_2)$.

В перечисленных интерпретациях основной задачи сюръективному отображению соответствуют такие размещения объектов по ящикам, что каждый ящик не пуст; раскраски объектов такие, что все цвета использованы при раскраске; слова в заданном алфавите, такие что в каждом слове использованы все буквы алфавита. Инъективному отображению соответствуют такие размещения объектов по ящикам, в которых каждый ящик содержит не более одного объекта; такие раскраски объектов, при которых цвета всех объектов различны и, наконец, слова в алфавите, все буквы которых различны.

Если x — действительное число, положим по определению

$$[x]_n = x(x-1)(x-2)\dots(x-n+1).$$

Обозначение $[x]_n$ читается как «n факториал от x вниз» или «нижняя n-ая степень x».

Утверждение 1.2. *Число инъективных отображений (инъекций) множества X из n элементов, $|X| = n$, во множество Y из m элементов, $|Y| = m$, есть $[m]_n$.*

Эквивалентное утверждение 1.2'. Число слов длины n без повторений букв в алфавите из m букв есть $[m]_n$.

Доказательство. Будем определять на этот раз число инъективных, (то есть имеющих все различные члены) последовательностей $< y_1, \dots, y_n >$. Элемент y_1 может быть выбран m способами, элемент y_2 можно выбрать $m-1$ способом из оставшихся элементов. В общем случае, если уже выбраны элементы $y_1, \dots, y_{i-1}$, то в качестве y_i может быть выбран любой из $m-i+1$ элементов множества Y $y_1, \dots, y_{i-1}$. (Принимаем, что $m \geq n$, если $n > m$, то и $[m]_n$ и искомое число функций равно 0). Это дает $m(m-1)\dots(m-n+1)$ возможность выбора инъективных последовательностей $< y_1, \dots, y_n >$. □

Отметим, что $[m]_n = m(m-1)\dots(m-n+1) = \dfrac{m(m-1)\dots 1}{(m-n)(m-n-1)\dots 1} = \dfrac{m!}{(m-n)!}$.

Определение. Каждое взаимно однозначное отображение $f : X \to X$ называется перестановкой множества X.

В соответствии с утверждением 2 число перестановок n-элементного множества равно $n!$.

1.3.1 Числа Стирлинга первого рода

Выражение $[x]_n$ является полиномом степени n от переменной x, следовательно его можно представить в виде следующего разложения по степеням x:

$$[x]_n = s(n,0) + s(n,1)x + \ldots + s(n,n)x_n$$

По определению, коэффициенты $s(n,k)$ такого разложения называются числами Стирлинга первого рода.

Утверждение 1.3. *Числа Стирлинга первого рода удовлетворяют следующим рекуррентным соотношениям:*

$$s(n+1,k) = s(n,k-1) - ns(n,k),\ k = 1,\ldots,n;\ s(n,0) = 0;\ s(n,n) = 1.$$

Доказательство. По определению $[x]_{n+1} = [x]_n(x-n)$. Представляя полиномы в левой и правой частях равенства в виде разложения по степеням x, получим: $s(n+1,n+1)x^{n+1}+\ldots+s(n+1,k)x^k+\ldots+s(n+1,0) = (s(n,n)+\ldots+s(n,k-1)x^{k-1}+s(n,k)x^k+\ldots+s(n,0))(x-n)$. Вычисляя и приравнивая коэффициенты при x^k слева и справа, получаем первую формулу утверждения. Другие две формулы очевидны. □

Утверждение 1.3 дает эффективный способ рекуррентного вычисления чисел Стирлинга первого рода.

Приведем часть значений таблицы $s(n,k)$ для начальных значений n и k.

$s(n,k)$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
$n=1$	0	1	0	0	0	0
$n=2$	0	-1	1	0	0	0
$n=3$	0	2	-3	1	0	0
$n=4$	0	-6	11	-6	1	0
$n=5$	0	24	-50	75	-10	1

Отметим связь чисел Стирлинга первого рода, в частности рекуррентного соотношения для них, с изучением циклической структуры перестановок.

1.3.2 Циклическая структура перестановок

Перестановки множеств и мультимножеств — один из самых богатых объектов перечислительной комбинаторики. Основная причина этого — большое разнообразие способов комбинаторного представления перестановки.

Перестановку можно представлять как слово или как функцию. В частности, функция $\pi : [n] \to [n]$, задаваемая равенством $\pi(i) = a_i$, соответствует слову $a_1 a_2 \dots a_n$.

Если рассматривать перестановку π конечного множества S как взаимно однозначное отображение $\pi : S \to S$, то естественно для каждого $x \in S$ рассмотреть последовательность $x,\ \pi(x),\ \pi^2(x),\ \dots$. В конце концов (так как π — взаимно однозначное соответствие, и множество S предполагается конечным) мы вновь получим x. Таким образом, для некоторого единственного наименьшего $k \geq 1$ имеем, что $\pi^k(x) = x$ и элементы $x,\ \pi(x),\ \dots,\ \pi^{k-1}(x)$ все различны. Назовем последовательность $(x,\ \pi(x),\ \dots,\ \pi^{k-1}(x))$ *циклом перестановки* π *длины* k. Циклы $(x,\ \pi(x),\ \dots,\ \pi^{k-1}(x))$ и $(\pi^i(x),\ \pi^{i+1}(x),\ \dots,\ \pi^{k-1}(x),\ x,\ \dots,\ \pi^{i-1}(x))$ считаются эквивалентными. Каждый элемент S встречается тогда в единственном цикле перестановки π, и мы можем рассматривать π как объединение непересекающихся циклов или, по-другому, как произведение различных циклов $C_1,\ \dots,\ C_n$.
Например, если перестановка $\pi : [7] \to [7]$ определена как

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 2 & 7 & 1 & 3 & 6 & 5 \end{pmatrix},$$

то есть $\pi(1) = 4,\ \pi(2) = 2,\ \pi(3) = 7,\ \pi(4) = 1,\ \pi(5) = 3,\ \pi(6) = 6,\ \pi(7) = 5$, то $\pi = (14)(2)(375)(6)$. Конечно, возможны различные обозначения такого представления перестановки; например, имеем: $\pi = (753)(14)(6)(2)$.
Можно определить стандартное представление:

- в каждом цикле пишется первым его наибольший элемент;
- циклы записываются в порядке возрастания их максимальных элементов.

Пусть $c(n,\ k)$ — число таких перестановок множества из n элементов, которые имеют k циклов. Будем обозначать множество всех перестановок n-элементного множества символом σ_n.

Утверждение 1.4. *Числа* $c(n,\ k)$ *удовлетворяют следующему рекуррентному соотношению:*

$$c(n,\ k) = (n-1)c(n-1,\ k) + c(n-1,\ k-1),\ n,\ k \geq 1$$

с начальными условиями $c(n,\ k) = 0$, *при* $n \leq 0$ *или* $k \leq 0$, *за исключением* $c(0,\ 0) = 1$.

Доказательство. Возьмем перестановку $\pi \in \sigma_{n-1}$ с k циклами. Мы можем вставить символ n после любого из символов $1,\ 2,\ \ldots,\ n-1$ в разложении перестановки на непересекающиеся циклы $n-1$ способом, получив таким образом разложение на непересекающиеся циклы перестановки $\pi' \in \sigma_n$ с k циклами, где n встречается в цикле длины, не меньшей 2. Следовательно, существует $(n-1)c(n-1,k)$ перестановок $\pi' \in \sigma_n$ с k циклами, для которых $\pi'(n) \neq n$.

С другой стороны, если выбрана перестановка $\pi' \in \sigma_{n-1}$ с $k-1$ циклом, ее можно достроить до перестановки $\pi' \in \sigma_n$ с k циклами, удовлетворяющей условию $\pi'(n) = n$. Положим

$$\pi'(i) = \begin{cases} \pi(i), & \text{если } i \in [n-1]; \\ n, & \text{если } i = n. \end{cases}$$

Следовательно, имеется $c(n-1,\ k-1)$ перестановок $\pi' \in \sigma_n$ с k циклами, для которых $\pi'(n) = n$, и доказательство закончено. □

Числа $c(n,\ k) = (-1)^{n-k}s(n,\ k)$ известны под названием *чисел Стирлинга первого рода без знака.*

Укажем на еще одну важную роль чисел $c(n,\ k)$.

Пусть x — переменная. Фиксируем $n \geq 0$. Тогда имеет место

Утверждение 1.5. $\sum\limits_{k=0}^{n} c(n,\ k)x^k = x(x+1)(x+2)\ldots(x+n-1).$

Доказательство. Положим

$$F_n(x) = (x+n-1)F_{n-1}(x) = \sum_{k=1}^{n} b(n-1,\ k-1)x^k + (n-1)\sum_{k=0}^{n-1} b(n-1,\ k)x^k.$$

Отсюда следует, что $b(n,\ k) = (n-1)b(n-1,\ k) + b(n-1,\ k-1)$. Поэтому $b(n,k)$ удовлетворяют тем же рекуррентным соотношениям и начальным условиям, что и $c(n,\ k)$, а значит, они совпадают. □

1.3.3 Упорядоченные размещения

Пусть x — переменная или действительное число.
Положим, по определению

$$[x]^n = x(x+1)(x+2)\ldots(x+n-1).$$

Обозначение $[x]^n$ читается как «n факториал от x вверх» или «верхняя n-ая степень x».

Определение. Пусть X — множество из n объектов $1, 2, \ldots, n$, которые должны быть размещены по m ящикам так, чтобы каждый ящик содержал бы последовательность, а не множество, как прежде, помещенных в нем объектов. Два размещения совпадают (равны), если в каждом ящике содержится одна и та же последовательность объектов. Размещения такого типа называются *упорядоченными размещениями n объектов по m ящикам.*

Приведем для примера всевозможные упорядоченные размещения двух объектов 1 и 2 в двух ящиках.

Ящики будем изображать в виде последовательности вертикальных черточек $|$, представляющих разделяющие ящики перегородки. Таким образом $2 \mid 1$ представляет размещение, при котором в первом ящике находится элемент 2, а во втором ящике — элемент 1.

Таблица всевозможных размещений двух объектов в двух ящиках имеет следующий вид:

$$\begin{array}{llll} \varnothing \mid 1\ 2; & 1 \mid 2; & 1\ 2 \mid \varnothing \\ \varnothing \mid 2\ 1; & 2 \mid 1; & 2\ 1 \mid \varnothing \end{array}$$

Утверждение 1.6. *Число упорядоченных размещений n объектов по m ящикам равно:*

$$[m]^n = m(m+1)\ldots(m+n-1)$$

(полагаем $[m]^0 = 1$)

Доказательство. Построим сначала таблицу T_{n-1} всех упорядоченных размещений объектов $1, 2, \ldots, n-1$ по m ящикам. Каждое размещение

$$i_1 i_2 \ldots \mid i_k i_{k+1} \ldots \mid \ldots \mid \ldots \mid \ldots i_{n-1}$$

можно представить как последовательность $(n-1)+(m-1)$ символов, являющихся либо буквой i_j, либо вертикальной чертой $|$. Чтобы из этой последовательности получить последовательность, представляющую упорядоченное размещение n объектов в нее достаточно всеми возможными способами добавить символ n. Символ n можно добавить к этой последовательности $(n-1)+(m-1)+1$ способами, помещая его перед самым первым символом, между двумя любыми символами и после последнего символа.

Таким образом,

$$|T_n| = (m+n-1)|T_{n-1}| = (m+n-1)(m+n-2)\ldots(m+1)|T_1| = [m]^n.$$

Очевидно, что $|T_1| = m$. □

Отметим простые, часто используемые соотношения:

$$[m]_n = (m-n+1)[m]_{n-1}; \qquad [m]^n = (m+n-1)[m]^{n-1}$$
$$[m]_n = \frac{m!}{(m-n)!}; \qquad [m]^n = \frac{(m+n-1)!}{(m-1)!}$$
$$[m]^n = [m+n-1]_n; \qquad [m]^n = [m]^{n-1}(m+n-1)$$

Определение. Пусть A — алфавит (то есть конечное множество символов) со множеством букв $a_1, \ldots, a_m$, упорядоченных так, что

$$a_1 < a_2 < \ldots < a_m.$$

Слово $x_1 x_2 \ldots x_n$ длины n — монотонное, если

$$x_1 < x_2 < \ldots x_n.$$

Пример

Пусть $A = \{a,\ b,\ c,\ d\},\ a < b < c < d.$
Тогда монотонными будут, например, следующие слова:

$$aaa,\ aab,\ abc,\ aad,\ bcd,\ ddd.$$

(По несколько устаревшей терминологии, это комбинации с повторениями из m объектов, взятые по n штук).

Утверждение 1.7. *Число монотонных слов длины n в алфавите из m букв равно* $\dfrac{[m]^n}{n!}$.

Доказательство. Рассмотрим упорядоченное размещение n объектов $1,\ 2,\ \ldots,\ n$ по m ящикам $a_1,\ \ldots,\ a_m$ и пусть ему соответствует монотонное слово следующим образом:

$$\underbrace{3}_{a_1} \mid \underbrace{251}_{a_2} \mid \underbrace{87}_{a_3} \mid \ldots \mid \underbrace{64n}_{a_m} \Rightarrow a_1 a_2 a_2 a_2 a_3 \ldots a_m a_m a_m.$$

В соответствующем слове буква a_1 написана столько раз, сколько объектов в ящике a_1, затем буква a_2 столько раз, сколько объектов в ящике a_2, ... Каждому упорядоченному размещению n объектов соответствует единственное монотонное слово. Все монотонные слова таким образом могут быть получены. Монотонному слову, с другой стороны, соответствует ровно $n!$ различных упорядоченных размещений. Поэтому число монотонных слов есть $\dfrac{[m]^n}{n!}$. □

Приложение. (Задача Муавра). Найдем число способов представления целого положительного числа m как упорядоченной суммы n неотрицательных целых чисел: $m = u_1 + u_2 + \ldots + u_n$.

Два таких представления

$$m = u_1 + u_2 + \ldots + u_n$$

и

$$m = u'_1 + u'_2 + \ldots + u'_n$$

будем считать совпадающими тогда и только тогда, когда

$$u_1 = u'_1,\ u_2 = u'_2,\ \ldots,\ u_n = u'_n,$$

то есть когда совпадают слагаемые и порядок их следования.

Положим значение σ_k равным частичной сумме первых k членов последовательности $u_1,\ \ldots,\ u_k:\ \sigma_k = u_1 + u_2 + \ldots + n_k$. Каждому представлению m в виде суммы n слагаемых взаимно однозначно соответствует слово

$$\sigma_1\sigma_2\ldots\sigma_{n-1}, \text{где } 0 \le \sigma_1 \le \sigma_2 \le \ldots \le \sigma_{n-1} \le m.$$

Таким образом, количество представлений m в виде упорядоченной суммы неотрицательных целых слагаемых равно количеству монотонных слов $\sigma_1\sigma_2\ldots\sigma_{n-1}$ длины $n-1$ в алфавите из $m+1$ символа ($\sigma_i \in \{0,\ 1, \ldots, m\}$, $i = 1,\ \ldots,\ n-1$).

Число представлений равно

$$\frac{[m+1]^{n-1}}{(n-1)!} = \frac{(m+n-1)!}{m!(n-1!)}.$$

1.3.4 Сочетания и биномиальные коэффициенты

Простейшими комбинаторными объектами являются сочетания и биномиальные коэффициенты.

Пусть дано конечное множество X, содержащее n различных элементов. Нас интересует количество различных k-элементных подмножеств, которые можно образовать из элементов множества X. Два подмножества считаются различными, если они различаются хотя бы одним входящим в них элементом.

Такие подмножества называются сочетаниями из m элементов по k элементов и обозначаются $\begin{pmatrix} X \\ k \end{pmatrix}$, а их количество обозначается

C_m^k или $\left(\begin{array}{c} m \\ k \end{array} \right)$. Обозначение читается как «число сочетаний из m по k» или просто «из m по k».

Утверждение 1.8. *Число различных подмножеств из k элементов множества A, $|A| = m$ есть*

$$C_m^k = \left(\begin{array}{c} m \\ k \end{array} \right) = \frac{[m]_k}{k!} = \frac{m(m-1)\dots(m-k+1)}{1 \cdot 2 \cdot \dots \cdot k} = \frac{m!}{k!(m-k)!}$$

Доказательство. 1 способ.
Построим таблицу T всех строго возрастающих (монотонных, без повторений букв) слов длины k в алфавите A из m букв.

Пример.

Пусть множество A состоит из пяти различных элементов: $A = \{a,\ b,\ c,\ d,\ e\}$.

Положим $k = 3$.

Тогда таблица T всех строго возрастающих слов длины 3 в алфавите A имеет следующий вид:

abc	acd	ade
abd	ace	
abe		
bcd	bde	
bce		
cde		

Переставим буквы в каждом слове всеми возможными способами и обозначим получившуюся таблицу T' . T' — множество слов без повторения букв длины k в алфавите A.

В таблице T' нет пропусков: каждое слово длины k появится в таблице T'.

В таблице T' нет повторений: два слова из T' либо получены из одного слова T и тогда отличаются порядком букв, либо из разных слов T и тогда различаются буквами.

По утверждению 1.2:

$$|T'| = [m]_k.$$

Поэтому:

$$|T| = \frac{[m]_k}{k!}.$$

Таким образом, окончательно получаем:

$$C_m^k = \left(\begin{array}{c} m \\ k \end{array} \right) = \begin{cases} \dfrac{[m]_k}{k!}, & \text{если } k \neq 0,\ m \geq k; \\ 1, & \text{если } k = 0,\ m \geq k; \\ 0, & \text{если } m < k. \end{cases}$$

2 способ.

Определим множество $\left(\begin{array}{c} S \\ k \end{array} \right)$ (иногда обозначаемое как-нибудь иначе) как множество всех k-элементных подмножеств (или k-подмножеств) множества S и положим по определению $\left(\begin{array}{c} n \\ k \end{array} \right) = \left| \left(\begin{array}{c} S \\ k \end{array} \right) \right|$ (игнорируя прошлое использование символа $\left(\begin{array}{c} n \\ k \end{array} \right)$). Подсчитаем двумя способами число $N(n,k)$ способов, которыми можно выбрать k-подмножество T множества S, а затем линейно упорядочить его элементы. Множество T мы можем выбрать $\left(\begin{array}{c} n \\ k \end{array} \right)$ способами, а затем k способами выбрать первый по порядку элемент множества T, $k-1$ способом — второй элемент T и так далее. Таким образом,

$$N(n,k) = \left(\begin{array}{c} n \\ k \end{array} \right) k!.$$

С другой стороны, можно взять n способами любой элемент множества S в качестве первого, $n-1$ способом любой из оставшихся элементов в качестве второго и так далее, k-ый элемент можно выбрать из оставшихся $n-k+1$ способом. Следовательно,

$$N(n,k) = n(n-1)\dots(n-k+1).$$

Итак, мы дали комбинаторное доказательство того, что:

$$\left(\begin{array}{c} n \\ k \end{array} \right) k! = n(n-1)(n-2)\dots(n-k+1),$$

и, следовательно,

$$\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n(n-1)(n-2)\dots(n-k+1)}{k!}.$$

□

Прежде, чем двигаться дальше сделаем небольшое отступление, связанное с введением понятия производящих функций.

Производящие функции

Производящие функции неизменно и естественно появляются во всех разделах перечислительного комбинаторного анализа. Мы будем делать акцент на наиболее органичном применении производящих функций для получения и проверки комбинаторных тождеств, когда другие методы менее естественны или менее эффективны. Производящие функции часто применяются в качестве метода, альтернативного методу рекуррентных соотношений, в частности с их помощью выводятся взаимно обратные соотношения.

Определение. Путь задана последовательность $a_1, a_2, \ldots, a_n, \ldots$ (неважно, конечная или бесконечная). *Производящей функцией последовательности* $a_1, a_2, \ldots, a_n, \ldots$ называется функция $A(x) = \sum\limits_{n=0}^{\infty} a_n x^n$ При этом все рассматриваемые ряды в случае бесконечной последовательности считаются формально сходящимися (если эти ряды сходятся в какой-то области к функции $f(x)$), поскольку мы интересуемся не областью сходимости соответствующих рядов, а лишь соотношениями между коэффициентами таких рядов.

Например, из формулы:

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \ldots + x^n + \ldots$$

вытекает, что функция $\dfrac{1}{1-x}$ является производящей функцией для последовательности чисел $1, 1, 1, \ldots, 1, \ldots$

Возводя обе части последнего разложения в квадрат, получаем:

$$\frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \ldots + (n+1)x^n + \ldots ,$$

откуда следует, что для последовательности $1, 2, 3, \ldots, n, \ldots$ производящей функцией является функция $\dfrac{1}{(1-x)^2}$.

Нас будут интересовать производящие функции для последовательностей $a_0, a_1, \ldots, a_n, \ldots$, так или иначе связанных с комбинаторными задачами. С помощью производящих функций удается получать и исследовать самые разные свойства этих последовательностей.

Пусть $B(x) = \sum\limits_{n=0}^{\infty} b_n x^n$ — производящая функция последовательности $b_1, b_2, \ldots, b_n, \ldots$ и $C(x) = \sum\limits_{n=0}^{\infty} c_n x^n$ — производящая функция последова-

тельности $c_1, c_2, \ldots, c_n, \ldots$. Тогда из равенства $C(x) = A(x)B(x)$ имеем $c_0 = a_0b_0$; $c_1 = a_1b_0 + a_0b_1$; $c_2 = a_2b_0 + a_1b_1 + a_0b_2$ или в общем виде:

$$c_n = a_nb_0 + a_{n-1}b_1 + \ldots + a_0b_n = \sum_{k=0}^{n} a_{n-k}b_k.$$

В таком случае говорят, что последовательность коэффициентов c_n есть свертка (произведение Коши) последовательностей a_n и b_n.

Биномиальные коэффициенты

Своё название биномиальные коэффициенты получили от соответствующей им производящей функции, являющейся степенью бинома:

$$(1+x)^m = \sum_{k=0}^{m} \left(\begin{array}{c} m \\ k \end{array} \right) x^k. \tag{1.1}$$

Для доказательства справедливости написанного соотношения (1.1) достаточно заметить, что коэффициент при x^k равен числу способов, которыми из m сомножителей $(1+x)\ldots(1+x)$ можно выбрать k сомножителей.

Отметим некоторые важнейшие соотношения для биномиальных коэффициентов (чисел сочетаний).

1.

$$C_n^k = C_n^{n-k} \tag{1.2}$$

Это важнейшее соотношение — прямое следствие того факта, что каждому k-элементному подмножеству $Y \subseteq X$ однозначно соответствует $(n-k)$ - элементное подмножество $X \backslash Y$ множества X.

2.

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1} \tag{1.3}$$

Зафиксируем некоторый элемент x из n-элементного множества X. Множество T всех k-элементных подмножеств множества X распадается на два непересекающихся класса:

$$T = T_1 \cup T_2;\ T_1 \cap T_2 = \varnothing,$$

класс T_1 подмножеств, которые не содержат элемент x, и класс T_2 подмножеств, которые его содержат. Мощность первого класса составляет C_{n-1}^k, а второго — C_{n-1}^{k-1}, то есть столько, сколько имеется $(k-1)$-элементных подмножеств множества $X \backslash \{x\}$. $\square$

Продемонстрируем эффективность использования производящей функции биномиальных коэффициентов для получения комбинаторных соотношений, включающих число сочетаний.

3. Полагая в (1.1) $x = 1$ получим:

$$\sum_{k=0}^{n} C_n^k = 2^n$$

Эта формула следует и из того, что сумма слева есть число всех подмножеств n-элементного множества.

4.

$$\sum_{k=0}^{n} kC_n^k = n2^{n-1}. \tag{1.4}$$

Дифференцируя (1.1) и полагая x=1, получаем соотношение (1.4).

5.

$$\left(\begin{array}{c} m+n \\ k \end{array} \right) = \sum_{s=0}^{k} \left(\begin{array}{c} m \\ s \end{array} \right) \left(\begin{array}{c} n \\ k-s \end{array} \right) \tag{1.5}$$

Равенство легко следует из следующего равенства для производящих функций:

$$(1+x)^{m+n} = (1+x)^m (1+x)^n.$$

Полагая в (1.5) $m = k = n$, получим:

$$C_{2n}^n = \sum_{r=0}^{n} \left(\begin{array}{c} n \\ r \end{array} \right)^2$$

Отметим, что задача прямого доказательства последнего равенства без использования производящей функции достаточно трудна.

6. Полагая в (1.1) $x = -1$, получаем

$$\sum_{k=0}^{n} (-1)^k \left(\begin{array}{c} m \\ k \end{array} \right) = 0$$

Отсюда следует, что

$$\sum_{k=0}^{[m/2]} \left(\begin{array}{c} m \\ 2k \end{array} \right) = \sum_{k=0}^{[m/2]} \left(\begin{array}{c} m \\ 2k+1 \end{array} \right) = 2^{m-1},$$

где через $[m/2]$ обозначена целая часть числа $m/2$.

Исчисление конечных разностей

Приведем пример использования биномиальных коэффициентов в вычислительной математике.

Пусть дана функция φ, определенная на множестве действительных (возможно целых) чисел и принимающая действительные значения. Определим новую функцию $\Delta\varphi(x)$, называемую первой разностью φ, формулой

$$\Delta\varphi(x) = \varphi(x+1) - \varphi(x).$$

Оператор Δ называется разностным оператором первого порядка; кратко и очень упрощенно можно определить исчисление конечных разностей как исследование оператора Δ. Можно применить оператор Δ k раз и получить k-ый разностный оператор

$$\Delta^k\varphi(x) = \Delta(\Delta^{k-1}\varphi(x)).$$

Число $\Delta^k(x)$ называется k-ой разностью φ в точке x ($\Delta^k\varphi(0)$ называется k-ой разностью φ в 0).

Определим другой оператор E, называемый оператором сдвига, формулой:

$$E\varphi(x) = \varphi(x+1).$$

Таким образом, $\Delta = E - I$, где I означает единичный оператор:

$$I\varphi(x) = \varphi(x).$$

Тогда первая разность функции может быть записана в виде:

$$\Delta\varphi(x) = \varphi(x+1) - \varphi(x) = E\varphi(x) - I\varphi(x) = (E-I)\varphi(x)$$

Разности более высоких порядков определяются рекуррентным соотношением: $\Delta^n\varphi(x) = \Delta(\Delta^{n-1}\varphi(x)) = \Delta^{n-1}\varphi(x+1) - \Delta^{n-1}\varphi(x) = (E-I)((E-I)^{n-1}\varphi(x))$. Откуда получаем выражение для n-ой разности:

$$\Delta^n\varphi(x) = (E-I)^n\varphi(x) = \sum_{k=0}^{n}(-1)^{n-k}C_n^kE^k\varphi(x) = \sum_{k=0}^{n}(-1)^{n-k}C_n^k\varphi(x+k)$$

В частности,

$$\Delta^k\varphi(0) = \sum_{i=0}^{k}(-1)^{k-i}\left(\begin{array}{c} k \\ i \end{array}\right)\varphi(i) \tag{1.6}$$

что даёт явную формулу для k-ой разности в терминах значений $\varphi(0)$, $\varphi(1)$, ..., $\varphi(k)$. Нетрудно обратить формулу (1.6) и выразить $\varphi(n)$ через $\Delta^i\varphi(0)$. Именно,

$$\varphi(n) = E^n\varphi(0) = (\Delta + 1)^n\varphi(0) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) \varphi(0) \tag{1.7}$$

Напишем теперь в строку значения:

$$\ldots \varphi(-2)\ \varphi(-1)\ \varphi(0)\ \varphi(1)\ \varphi(2) \ldots$$

Если внизу написать между каждой парой последовательных членов $\varphi(i)$, $\varphi(i+1)$ их разность $\varphi(i+1) - \varphi(i) = \Delta\varphi(i)$, то получим последовательность:

$$\ldots \Delta\varphi(-2)\ \Delta\varphi(-1)\ \Delta\varphi(0)\ \Delta\varphi(1)\ \Delta\varphi(2) \ldots$$

Повторение этой процедуры приводит к таблице разностей функции φ , k-ая строка которой состоит из значений $\Delta^k\varphi(n)$. Диагональ, начинающаяся в $\varphi(0)$ и идущая направо вниз, состоит из разностей $\Delta^k\varphi(0)$ в 0. Например, пусть $\varphi(n) = n4$. Таблица разностей (начинающаяся с $\varphi(0)$) выглядит так:

$$\begin{array}{ccccccccccc}
0 & & 1 & & 16 & & 81 & & 256 & & 625 \quad \ldots \\
& 1 & & 15 & & 65 & & 175 & & 369 & \\
& & 14 & & 50 & & 110 & & 194 & & \\
& & & 36 & & 60 & & 84 & & & \\
& & & & 24 & & 24 & & & & \\
& & & & & 0 & & & & &
\end{array}$$

Из формулы (1.7) следует, что:

$$n^4 = \left(\begin{array}{c} n \\ 1 \end{array} \right) + 14 \left(\begin{array}{c} n \\ 2 \end{array} \right) + 36 \left(\begin{array}{c} n \\ 3 \end{array} \right) + 24 \left(\begin{array}{c} n \\ 4 \end{array} \right) + 0 \left(\begin{array}{c} n \\ 5 \end{array} \right) + \ldots \tag{1.8}$$

В этом случае, так как n^4 — многочлен четвертой степени и $\left(\begin{array}{c} n \\ k \end{array} \right)$ при фиксированном k есть многочлен степени k, написанное выше разложение обрывается после члена $24 \left(\begin{array}{c} n \\ 4 \end{array} \right)$, то есть $\Delta^k 0^4 = 0$, если $k > 4$ (или, более общим образом, $\Delta^k n^k = 0$, если $k > 4$).

Предыдущее рассуждение, конечно, не относится лишь к функции n^4. Подобные рассуждения приводят к следующим результатам.

1. Функция φ — полином степени, не превосходящей d, тогда и только тогда, когда $\Delta^{d+1}\varphi(n) = 0$ (или $\Delta^d\varphi(n)$ — постоянная).

2. Если многочлен $\varphi(n)$ степени, не превосходящей d, разложен в ряд по базису $\binom{n}{k}$, $0 \le k \le d$, то коэффициенты разложения есть $\Delta^k\varphi(0)$, то есть

$$\varphi(n) = \sum_{k=0}^{n} \Delta^k\varphi(0) \binom{n}{k}. \quad \square$$

Еще одна связь комбинаторных объектов с исчислением конечных разностей дается формулой (1.15).

Разложения

Существует тесная связь между подмножествами множеств и разложениями целого числа.

Определение. Разложение n есть представление числа n в виде упорядоченной суммы положительных целых. Например, существует восемь разложений числа 4, а именно:

$1+1+1+1$	$3+1$
$2+1+1$	$1+3$
$1+2+1$	$2+2$
$1+1+2$	4

Если разложение σ содержит в точности k слагаемых, то говорят, что σ имеет k частей и называется k-разложением. Если $a_1 + a_2 + \ldots + a_k$ — k -разложение σ числа n, определим $(k-1)$-элементное подмножество $\Theta(\sigma)$ (или $(k-1)$-подмножество) множества $\{1,\ 2,\ \ldots,\ n-1\}$ формулой:

$$\Theta(\sigma) = \{a_1,\ a_1 + a_2,\ \ldots,\ a_1 + a_2 + \ldots + a_{k-1}\}.$$

Эта формула устанавливает взаимно однозначное соответствие (биекцию) между всеми k-разложениями и $(k-1)$-подмножествами $\Theta(\sigma)$ множества $\{1,\ 2,\ \ldots,\ n-1\}$. Следовательно, существует $\binom{n-1}{k-1}$ k-разложений n и 2^{n-1} разложений n. Разложение часто схематично представляют, рисуя в строку n точек и $k-1$ разделяющую вертикальную черту. Точки разделяются по k линейно упорядоченным «отделениям», числа точек в

отделениях дают k-разложение числа n. Например, последовательность

$$\bullet \mid \bullet\bullet \mid \bullet \mid \bullet \mid \bullet\bullet\bullet \mid \bullet\bullet$$

соответствует разложению $1+2+1+1+3+2$.

Другая проблема, тесно связанная с разложениями, есть задача подсчета числа $N(n,k)$ решений уравнения

$$x_1 + x_2 + \ldots + x_k = n$$

в *неотрицательных* целых числах. Решение такого уравнения называется *слабым* разложением n на k частей, или слабым k-разложением числа n. (Решение в положительных целых числах есть просто k-разложение n.) Если мы положим $y_1 = x_1 + 1$, $y_2 = x_2 + 1$, $\ldots$, $y_k = x_k + 1$, то получим, что $N(n,\ k)$ есть количество решений в положительных числах уравнения

$$y_1 + y_2 + \ldots + y_k = n + k,$$

то есть число k-разложений числа $n+k$.

Таким образом, $N(n,\ k) = \left(\begin{array}{c} n+k-1 \\ k-1 \end{array} \right)$.

Подобным же приемом (найти его предоставляется читателю) доказывается, что число решений неравенства $x_1 + x_2 + \ldots + x_k \leq n$ в неотрицательных целых числах есть $\left(\begin{array}{c} n+k \\ k \end{array} \right)$.

k-подмножество T n множества S иногда называют k-сочетанием из S без повторений. Так возникает задача подсчета числа k-сочетаний из S с повторениями; то есть мы выбираем k элементов из множества S, не взирая на их порядок и допуская повторяющиеся элементы. Обозначим число таких способов $\left(\left(\begin{array}{c} n+k \\ k \end{array} \right) \right)$. Например, $\left(\left(\begin{array}{c} 3 \\ 2 \end{array} \right) \right) = 6$.

Если $S = \{1,\ 2,\ 3\}$, то подходящие сочетания есть 11, 22, 33, 12, 13 и 23. Эквивалентное, но более точное определение и исследование сочетаний с повторениями может быть проведено, если ввести понятие мультимножества. На интуитивном уровне мультимножество есть множество с повторяющимися элементами, например $\{1,\ 1,\ 2,\ 5,\ 5\}$. Более точно, конечное мультимножество M на множестве S есть функция $\nu : S \to \mathbb{N}$ ($\mathbb{N}$ - множество натуральных чисел), такая, что $\sum\limits_{x \in S} \nu(x) < \infty$ рассматривается как число повторений элемента x. Целое число $\sum\limits_{x \in S} \nu(x)$ называют мощностью или числом элементов M и обозначают $|M|$. Если $S = \{x_1,\ x_2,\ \ldots,\ x_n\}$,

и $\nu(x_i) = a_i$, то мы пишем $M = \{x_i^{a_i}, \ldots, x_n^{a_n}\}$. Множество всех k-мультимножеств на S обозначается символом $\left(\!\left(\begin{array}{c} S \\ k \end{array}\right)\!\right)$.

Если $S = \{y_1, \ldots, y_n\}$ и мы положим $x_i = \nu(y_i)$, то увидим, что $\left(\!\left(\begin{array}{c} n \\ k \end{array}\right)\!\right)$. есть число решений в неотрицательных целых числах уравнения $x_1 + x_2 + \ldots + x_n = k$. Это число, как мы видели, есть

$$\left(\begin{array}{c} n+k-1 \\ n-1 \end{array}\right). \tag{1.9}$$

Прямое комбинаторное доказательство утверждения (1.9) таково. Пусть $\{a_1, a_2, \ldots, a_k\}$, $1 \le a1 < a2 < \ldots < a_k \le n+k-1$, есть k-подмножество множества $[n+k-1] = \{1, 2, \ldots, n+k-1\}$. Положим $b_i = a_i - i + 1$. Тогда $\{b_1, b_2, \ldots, b_n\}$ — k-мультимножество на множестве $[n]$.

Обратно, если дано k-мультимножество $1 \le b_1 \le b_2 \le b_k \le n$ на $[n]$, то определив a_i формулой $a_i = b_i + i - 1$, видим, что $\{a_1, a_2, \ldots, a\}$ есть k-подмножество множества $[n+k-1]$. Следовательно, мы определили взаимно однозначное соответствие между $\left(\!\left(\begin{array}{c} [n] \\ k \end{array}\right)\!\right)$ и $\left(\!\left(\begin{array}{c} [n+k-1] \\ k \end{array}\right)\!\right)$, что и требовалось доказать.

Поучителен подход к мультимножествам с точки зрения производящих функций. Совершенно аналогично проведенному исследованию подмножеств множества $S = \{x_1, x_2, \ldots, x_n\}$ имеем

$$(1 + x_1 + x_1^2 + \ldots)(1 + x_2 + x_2^2 + \ldots)\ldots(1 + x_n + x_n^2 + \ldots) = \sum_{\nu: S \to \mathbb{N}} \prod_{x_i \in S} x_i^{\nu(x_i)}$$

Положим $x_i = x$. Тогда

$$(1 + x + x^2 + \ldots)^n = \sum_{\nu} x^{\nu(x_1)+\ldots+\nu(x_n)} = \sum_{\text{M на S}} x^{|M|} = \sum_{k \ge 0} \left(\!\left(\begin{array}{c} n \\ k \end{array}\right)\!\right) x^k.$$

Но,

$$(1 + x + x^2 + \ldots)^n = (1 - x)^{-n} = \sum_{k \ge 0} \left(\begin{array}{c} -n \\ k \end{array}\right) (-1)^k x^k,$$

так что

$$\left(\begin{array}{c} n \\ k \end{array}\right) = (-1)^k \left(\begin{array}{c} -n \\ k \end{array}\right) = \left(\begin{array}{c} n+k-1 \\ k \end{array}\right)$$

Появление элегантной формулы

$$\left(\begin{array}{c} n \\ k \end{array} \right) = (-1)^k \left(\begin{array}{c} -n \\ k \end{array} \right)$$

не случайно. Это простейший пример комбинаторной теории взаимности.

1.3.5 Полиномиальные коэффициенты

Утверждение 1.9. *Пусть X множество n различных объектов и пусть $n_1, n_2, \ldots, n_p$ неотрицательные целые числа, удовлетворяющие условию $n_1 + n_2 + \ldots + n_p = n$; количество размещений n объектов по ячейкам $Y_1, \ldots, Y_p$, при которых каждая ячейка содержит $n_1, n_2, \ldots, n_p$ объектов соответственно, есть*

$$\left(\begin{array}{c} n \\ n_1, n_2, \ldots, n_p \end{array} \right) = \begin{cases} \dfrac{n!}{n_1! n_2! \ldots n_p!}, & \text{если } n_1 + n_2 + \ldots + n_p = n, \\ 0, & \text{если } n_1 + n_2 + \ldots + n_p \neq n. \end{cases}$$

Доказательство. Пусть $n_1 + n_2 + \ldots + n_p = n$. Ящик Y_1 можно наполнить $\left(\begin{array}{c} n \\ n_1 \end{array} \right)$ различными способами, после чего ящик Y_2 можно наполнить $\left(\begin{array}{c} n - n_1 \\ n_2 \end{array} \right)$ способами и так далее.
Следовательно, искомое число размещений равно:

$$\left(\begin{array}{c} n \\ n_1, n_2, \ldots, n_p \end{array} \right) = \left(\begin{array}{c} n \\ n_1 \end{array} \right) \left(\begin{array}{c} n - n_1 \\ n_2 \end{array} \right) \left(\begin{array}{c} n - n_1 - n_2 \\ n_3 \end{array} \right) \ldots \left(\begin{array}{c} n_p \\ n_p \end{array} \right) =$$
$$= \frac{n!}{n_1!(n - n_1)!} \cdot \frac{(n - n_1)!}{n_2!(n - n_1 - n_2)!} \cdot$$
$$\cdot \frac{(n - n_1 - n_2)!}{n_3!(n - n_1 - n_2 - n_3)!} \cdot \ldots \cdot \frac{n_p!}{n_p!} =$$
$$= \frac{n!}{n_1! n_2! n_3! \ldots n_p!}.$$

□

Утверждение 1.10. *Производящая функция для полиномиальных коэффициентов имеет следующий вид:*

$$(x_1+x_2+\ldots+x_n)^n = \sum_{\substack{n_1,\,n_2,\,\ldots,\,n_p\geq 0\\ n_1+n_2+\ldots+n_p=n}} \left(\begin{array}{c} n \\ n_1, n_2, \ldots, n_p \end{array} \right) x_1^{n_1} x_2^{n_2} \ldots x_p^{n_p} \quad (1.10)$$

Доказательство. Для доказательства справедливости равенства (1.10) достаточно заметить, что коэффициент при $x_1^{n_1} x_2^{n_2} \ldots x_p^{n_p}$ равен числу способов выбрать из n сомножителей n_1 сомножителей, из которых в произведение войдет переменная x_1, n_2 сомножителей, из которых в произведение войдет переменная x_2, и так далее. □

Следствие 1.

$$\left(\begin{array}{c} n \\ n_1,\ n_2,\ \ldots,\ n_p \end{array} \right) = \sum_{\substack{j\\ n_j\neq 0}} \left(\begin{array}{c} n-1 \\ n_1,\ n_2,\ \ldots,\ n_{j-1},\ n_j-1,\ n_{j+1},\ \ldots,\ n_p \end{array} \right) \quad (1.11)$$

Для доказательства следствия 1 достаточно заметить, что

$$(x_1 + x_2 + \ldots + x_n)^n = (x_1 + x_2 + \ldots + x_n)^{n-1}(x_1 + x_2 + \ldots + x_n).$$

Отсюда следует равенство коэффициентов при соответствующих степенях в левой и правой частях последнего равенства:

$$x_1^{n_1} \ldots x_p^{n_p} \left(\begin{array}{c} n \\ n_1,\ \ldots,\ n_p \end{array} \right) =$$

$$= \sum_{\substack{j\\ n_j\neq 0}} x_j \left(\begin{array}{c} n-1 \\ n_1,\ \ldots,\ n_{j-1}, n_j-1, n_{j+1},\ \ldots,\ n_p \end{array} \right) x_1^{n_1} x_2^{n_2} \ldots x_j^{n_j-1} \ldots x_n^{n_p}. \ \square$$

Следствие 2

$$\left(\begin{array}{c} m+q \\ n_1, n_2,\ \ldots,\ n_p \end{array} \right) = \sum_{\substack{(k_1,k_2,\,\ldots,\,k_p)\leq(n_1,n_2,\,\ldots,\,n_p)\\ k_1+k_2+\ldots=m}} \left(\begin{array}{c} m \\ k_1,\ \ldots,\ k_p \end{array} \right) \cdot$$

$$\cdot \left(\begin{array}{c} q \\ n_1-k_1, n_2-k_2,\ \ldots,\ n_p \end{array} \right)$$

Указанное равенство есть непосредственное следствие следующего соотношения для производящих функций:

$$(x_1 + \ldots + x_p)^{m+q} = (x_1 + \ldots + x_p)^m (x_1 + \ldots + x_p)^q. \ \square$$

Следствие 3

$$\sum \left(\begin{array}{c} n \\ n_1,\ \ldots,\ n_p \end{array} \right) (-1)^{n_2+n_4+n_6+\ldots} = \frac{1-(-1)^p}{2}$$

Равенство следствия 3 непосредственно вытекает из вида производящей функции для полиномиальных коэффициентов, если в (1.10) положить:

$$x_1 = x_3 = x_5 = \ldots = +1$$
$$x_2 = x_4 = x_6 = \ldots = -1 \qquad \square$$

1.4 Разбиения

1.4.1 Число разбиений

Определение. *Разбиение* конечного множества X,$|X| = n$, есть неупорядоченный набор $\pi = \{B_1,\ B_2,\ \ldots,\ B_k\}$ подмножеств множества X таких, что

$B_i \neq \varnothing$ для всех i от 1 до k;
$B_i \cap B_j = \varnothing$, если $i \neq j$
$B_1 \cup B_2 \cup \ldots \cup B_k = X$

Мы называем B_i *классом (блоком)* разбиения π и говорим, что π имеет k классов. Пусть $S(n,k)$ — число разбиений n - множества на k классов.

$S(n,k)$ называется также числом Стирлинга второго рода. $\square$

Разбиения соответствуют размещениям n различных объектов по k одинаковым ящикам при условии, что каждый ящик не пуст.

Пример

$S(4,\ 2) = 7$. Действительно, четыре объекта $\{1,\ 2,\ 3,\ 4\}$ можно следующим образом разбить на два класса:

$\{1\} \cup \{2,\ 3,\ 4\}$; $\{3\} \cup \{1,\ 2,\ 4\}$ $\{1,\ 2\} \cup \{3,\ 4\}$ $\{1,\ 4\} \cup \{2,\ 3\}$
$\{2\} \cup \{1,\ 3,\ 4\}$ $\{4\} \cup \{1,\ 2,\ 3\}$ $\{1,\ 3\} \cup \{2,\ 4\}$

Условимся полагать, что $S(0,\ 0) = 1$.

Читатель должен убедиться, что для $n \geq 1$ имеют место соотношения:

$S(0,\ k) = 0$, при $k > 0$,
$S(n,\ k) = 0$, при $k > n$,
$S(n,\ 0) = 0$,
$S(n,\ 1) = 1$,
$S(n,\ 2) = 2^{n-1} - 1$,
$S(n,\ n) = 1$,
$S(n,\ n-1) = \left(\begin{array}{c} n \\ 2 \end{array} \right)$.

Утверждение 1.11. *Числа Стирлинга второго рода удовлетворяют следующему основному рекуррентному соотношению:*

$$S(n+1,\ k) = S(n,\ k-1) + kS(n,\ k). \tag{1.12}$$

Доказательство. Рассмотрим таблицу разбиений $n+1$ объекта на k классов.

1. Для некоторых разбиений $(n+1)$-ый объект есть единственный элемент в классе. Число таких разбиений есть $S(n,\ k-1)$.

2. Для других разбиений $(n+1)$-ый объект не является единственным элементом класса ни для какого класса. Следовательно, существует $kS(n,\ k)$ таких разбиений, так как каждому разбиению множества $\{1,\ \ldots,\ n\}$ на k классов соответствует в точности k разбиений, образованных добавлением элемента $n+1$ поочередно к каждому классу.

Таким образом, мы представили все разбиения $n+1$ элемента на k классов в виде объединения непересекающихся подмножеств разбиений двух перечисленных типов. Поэтому

$$S(n+1,\ k) = S(n,\ k-1) + kS(n,\ k).$$

□

Утверждение 1.12. *Число сюръективных отображений множества X, $|X| = n$, на множество $Y(|Y| = m)$ равно $m!\,S(n,m)$.*

Доказательство. Каждое сюръективное отображение $X = \{1,\ 2,\ \ldots,\ n\}$ на $Y = \{y_1,\ y_2,\ \ldots,\ y_m\}$ индуцирует разбиение X на m различных классов $1,\ 2,\ \ldots,\ m$ (в класс i попадают все такие x, что $f(x) = y_i$); наоборот, каждому разбиению X на m классов соответствует $m!$ сюръективных отображений X на Y. Действительно, выражение $m!\,S(n,\ m)$ дает число способов разбить X на m классов, а затем линейно упорядочить классы, скажем, $(B_1,\ B_2,\ \ldots,\ B_m)$. Свяжем последовательность $(B_1,\ B_2,\ \ldots,\ B_m)$ с сюръективной функцией f, определенной формулой $f(i) = y_j$, если $i \in B_j$. Это устанавливает требуемое соответствие между количеством сюръективных отображений и числом разбиений. □

Ниже приводится список некоторых основных формул для количества разбиений множества из n элементов на k классов — $S(n,\ k)$.

1. Формула 1.

$$x^n = \sum_{k=0}^{n} S(n,\ k)[x]_k \tag{1.13}$$

Числа $S(n,\ k)$ играют обратную роль по отношению к числам $s(n,\ k)$ — позволяют перейти от базиса $1,\ x,\ x^2,\ \ldots$ к базису $[x]_1,\ [x]_2,\ \ldots$

Доказательство. Рассмотрим всевозможные отображения множества X из n элементов ($|X| = n$) во множество Y из m элементов ($|Y| = m$). С одной стороны, по утверждению 1.1 количество таких отображений есть m^n. С другой стороны, каждое такое отображение есть сюръективное отображение множества X на подмножество $B \subseteq Y$. Для произвольного подмножества $B \subseteq Y$, где $|B| = k \leq n$ число сюръективных функций $f : X \to B$ в соответствии с утверждением 1.12 равно $k!\,S(n,\ k)$. Учитывая, что подмножество B мощности k можно выбрать C_m^k способами получаем формулу:

$$m^n = \sum_{k=1}^{m} C_m^k k! S(n,\ k) = C_m^1 1! S(n,\ 1) + \ldots + C_m^n n! S(n,\ n) \tag{1.14}$$

Равенство (1.14) можно рассматривать как равенство двух многочленов переменной x при всех целых положительных значениях $x = m$. Следовательно, эти многочлены тождественно равны между собой, так как их разность может быть либо тождественным нулем, либо должна иметь бесконечное число нулей, что невозможно. Справедливость формулы (1.13) доказана. □

2. Формула 2.

$$S(n+1,\ m) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) S(k,\ m-1) = \sum_{k=m-1}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) S(k,\ m-1)$$

Доказательство. Рассмотрим множество всех разбиений множества $X = \{1,\ 2,\ \ldots,\ n+1\}$ на m классов. Количество таких разбиений есть $S(n+1,\ m)$. Все разбиения распадаются на различные типы, соответствующие разным подмножествам множества X, содержащим элемент $n+1$. Для каждого k-элементного подмножества $B \subset X$, содержащего элемент $n+1$, существует в точности $S(n+1-k,\ m-1)$ разбиений множества X на $m-1$ класс, содержащих B в качестве класса. Действительно, каждое такое разбиение однозначно соответствует разбиению множества $X \setminus B$ на

$m-1$ класс. k-элементное подмножество $B \subset X$, содержащее элемент $n+1$ можно выбрать $\left(\begin{array}{c} n \\ k-1 \end{array} \right)$ способами. Таким образом, имеем:

$$S(n+1,\ m) = \sum_{k=1}^{1+n-(m-1)} \left(\begin{array}{c} n \\ k-1 \end{array} \right) S(n+1-k,\ m-1) =$$

$$= \sum_{k=1}^{1+n-(m-1)} \left(\begin{array}{c} n \\ n-k-1 \end{array} \right) S(n+1-k,\ m-1) =$$

$$= \sum_{r=m-1}^{n} \left(\begin{array}{c} n \\ r \end{array} \right) S(r,\ m-1) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ r \end{array} \right) S(k,\ m-1)$$

□

3. Вернемся еще раз к связи комбинаторных объектов с исчислением конечных разностей. Из формулы (1.13) следует, что, например,

$$n^4 = \sum_{k=0}^{4} k! S(4,\ k) \left(\begin{array}{c} n \\ k \end{array} \right), \tag{1.15}$$

откуда заключаем на основании разложения (1.8):

$$1!S(4,\ 1) = 1,\ \ 2!S(4,\ 2) = 14,\ \ 3!S(4,\ 3) = 36,\ \ 4!S(4,\ 4) = 24.$$

Указанная связь дает альтернативный способ вычисления последовательности $S(n,\ k)$. □

1.4.2 Числа Белла

Определение. Общее число разбиений множества $X, |X| = n$ на произвольные классы называется *числом Белла* и обозначается $B(n)$. Таким образом по определению:

$$B(n) = \sum_{k=1}^{n} S(n,\ k),\ \ n \geq 1.$$

Положим по определению $B(0) = 1$.

Формула 3.

$$B(n+1) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) B(k).$$

Доказательство. Напомним, что $S(n,\ m) = 0$ при $m > n$.
Тогда имеем следующую последовательность очевидных равенств:

$$B(n+1) = \sum_{r=1}^{n+1} S(n+1,\ r) = \sum_{r=1}^{n+1} \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) S(k,\ r-1) =$$

$$= \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) \left(\sum_{r=1}^{n+1} S(k,\ r-1) \right) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) B(k)$$

□

1.5 Принцип включений-исключений

Этот раздел посвящен важному комбинаторному методу — принципу включений-исключений, известному также под названиями: символический метод, принцип перекрестной классификации, метод решета. Логические тождество, на котором основаны все эти методы, известны давно. Еще в 1713 году Монмор эффективно использовал упомянутый метод в решении знаменитой задачи о встречах (о числе перестановок из n элементов, в которых ни один элемент не сохраняет своей позиции).

Принцип включений-исключений — одно из фундаментальных средств перечислительной комбинаторики. Красота этого принципа лежит не в самом результате, а в его широкой применимости.

Принцип включений-исключений в перечислительной комбинаторике есть метод определения мощности множества S, который начинает с большего множества и каким-либо путем вычитает или аннулирует нежелательные элементы. Сначала дается приблизительный ответ, содержащий большее число элементов, затем вычитается число элементов, большее чем ошибка, полученная на первом шаге, пока мы не придем к правильному ответу. Это комбинаторная сущность принципа включения-исключения.

Для примера рассмотрим принцип включений-исключений в теоретико-множественной форме.

Пусть даны два конечных множества A и B.

Тогда:
$|A \cup B| = |A| + |B| - |A \cap B|$.

Чтобы вычислить количество элементов в объединении двух множеств, мы сначала вычисляем сумму их мощностей, но при этом дважды учитываем каждый элемент, принадлежащий пересечению множеств. Вычитая мощность пересечения, приходим к правильному ответу.

Совершенно аналогичные рассуждения позволяют выписать формулу для количества элементов в объединении трех множеств A, B, и C:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Вычитая дважды учтенные элементы попарных пересечений, мы трижды вычли элементы, принадлежащие пересечению всех трех множеств. Добавление мощности пересечения $A \cap B \cap C$ приводит к нужному результату.

Пусть имеется N объектов и n различных свойств $\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n$. Каждый из объектов может обладать любым из этих свойств (в любом наборе), т.е. обладать любым набором этих свойств, или не обладать никаким из свойств.

Пусть $N(\alpha_1)$ — число объектов обладающих свойством α_1. Некоторые из этих объектов могут обладать и другими свойствами в дополнение к α_1, но это неважно. (На самом деле в этом и состоит вся идея метода включений-исключений). Пусть теперь $N(\alpha_2)$ — число объектов, обладающих свойством α_2, и так далее. Соответственно, через $N(\alpha_1,\ \alpha_2)$ обозначим количество объектов, обладающих двумя свойствами: свойством α_1 и свойством α_2.

В общем случае пусть $N(\alpha_{i_1},\ \alpha_{i_2},\ \ldots,\ \alpha_{i_s})$ — число объектов, обладающих свойствами $\alpha_{i_1},\ \alpha_{i_2},\ \ldots,\ \alpha_{i_s}$ (и, быть может, некоторыми другими).

Пусть N_0 — число предметов, не обладающих ни одним из свойств $\alpha_1,\ \ldots,\ \alpha_n$.

Пусть $N(\overline{\alpha}_1)$ — число объектов, не обладающих свойством α_1. Чертой над символом свойства будем указывать, что речь идет об объектах, не обладающих таким свойством. Тогда в принятом обозначении $N_0 = N(\overline{\alpha}_1,\ \overline{\alpha}_2,\ \ldots,\ \overline{\alpha}_n)$.

Теорема 1.13 (Формула включений-исключений)**.**

$$N_0 = N - \sum_i N(\alpha_i) + \sum_{1 \le i < j \le n} N(\alpha_i,\ \alpha_j) -$$
$$- \sum_{1 \le i < j < k \le n} N(\alpha_i,\ \alpha_j,\ \alpha_k) + \ldots + (-1)^n N(\alpha_1,\ \ldots,\ \alpha_n) \qquad (1.16)$$

Прежде, чем переходить к доказательству, вернемся к ранее рассмотренному примеру с тремя множествами A, B, C. Пусть свойством α_1 обладают все элементы множества A, свойством α_2 обладают все элементы множества B, свойством α_3 — все элементы, принадлежащие множеству C. Тогда очевидно, что количество элементов, не обладающих ни одним из

свойств $\alpha_1,\ \alpha_2,\ \alpha_3$ равно 0 (каждый элемент принадлежит хотя бы одному из множеств), и в соответствии с формулой (1.16) имеем:

$$N_0 = 0 = |A \cup B \cup C| - |A| - |B| - |C| + |A \cap B| + |B \cap C| + |A \cap C| - |A \cap B \cap C|.$$

Доказательство. Доказательство проводится индукцией по n — числу свойств. При одном свойстве α формула очевидна. Каждый объект либо обладает этим свойством, либо не обладает им. Поэтому $N_0 = N - N(\alpha)$.

Предположим теперь, что для случая, когда число свойств равно $n-1$, формула доказана:

$$N_0 = N - N(\alpha_1) - \ldots - N(\alpha_{n-1}) + N(\alpha_1,\ \alpha_2) + \ldots$$
$$+ N(\alpha_{n-2},\ \alpha_{n-1}) + (-1)^{n-1} N(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_{n-1}). \qquad (1.17)$$

Отметим очевидное соотношение:

$$N(\overline{\alpha}_1,\ \overline{\alpha}_2,\ \ldots,\ \overline{\alpha}_n) = N(\overline{\alpha}_1,\ \ldots,\ \overline{\alpha}_{n-1}) - N(\overline{\alpha}_1,\ \ldots,\ \overline{\alpha}_{n-1}, \alpha_n) \qquad (1.18)$$

Формула (1.17) по предположению справедлива для любой совокупности объектов. В частности она верна для совокупности $N(\alpha_n)$ элементов, обладающих свойством α_n. Применим индуктивное предположение к совокупности $N(\alpha_n)$ для вычисления $N(\overline{\alpha}_1,\ \ldots,\ \overline{\alpha}_{n-1},\ \alpha_n)$:

$$N(\overline{\alpha}_1,\ \ldots,\ \overline{\alpha}_{n-1},\ \alpha_n) = N(\alpha_n) - \sum_{1 \le i \le n-1} N(\alpha_i,\ \alpha_n) +$$
$$+ \sum_{1 \le i < j \le n-1} N(\alpha_i,\ \alpha_j,\ \alpha_n) + \ldots + (-1)^{n-1} N(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n) \qquad (1.19)$$

Вычтем равенство (1.19) из (1.17). В правой части получим то, что нужно — правую часть формулы включения-исключения, а в левой части получим разность (1.18). Тем самым формула (1.16) доказана. □

Для того, чтобы облегчить всестороннее использование этой формулы, сделаем следующие замечания. Во-первых, лучше всего она запоминается в следующей «символической записи»:

$$N(\overline{\alpha}_1,\ \overline{\alpha}_2,\ \overline{\alpha}_3,\ \ldots) = N[(1-\alpha_1)(1-\alpha_2)(1-\alpha_3)\ldots]$$

Сначала вычисляется содержимое квадратных скобок, а затем знак функции N применяется к слагаемым:

$$N[\alpha + \beta] = N(\alpha) + N(\beta),$$
$$N[-\alpha] = -N[\alpha],$$
$$N[1] = N,$$
$$N[(1-\alpha)(1-\beta)] = N(1 - \alpha - \beta + \alpha\beta) = N - N(\alpha) - N(\beta) + N(\alpha,\ \beta).$$

Далее, как видно из доказательства формулы включений-исключений, совокупности объектов, к которым применима теорема, не обязана быть совокупностью N всех объектов. Поэтому:

$$N(\alpha_1\overline{\alpha}_2\alpha_3\overline{\alpha}_4) = N[\alpha_1(1-\alpha_2)\alpha_3(1-\alpha_4)] = N(\alpha_1\alpha_3) - N(\alpha_1\alpha_2\alpha_3) - \\ - N(\alpha_1\alpha_3\alpha_4) + N(\alpha_1\alpha_2\alpha_3\alpha_4)$$

Вообще, если n различных свойств $\alpha_1, \ldots, \alpha_n$ объектов из совокупности N объектов обозначить $b_1, b_2, \ldots, b_k, \bar{c}_1, \bar{c}_2, \ldots, \bar{c}_{n-k}$, то

$$N[b_1b_2\ldots b_k\bar{c}_1\bar{c}_2\ldots\bar{c}_{n-k}] = N[b_1b_2\ldots b_k(1-c_1)\ldots(1-c_{n-k})] \qquad \square$$

Рассмотрим принцип включений-исключений в несколько более общей форме. Пусть S — множество свойств, которыми элементы данного множества A могут обладать, а могут не обладать. Для любого подмножества T множества S, $T \subseteq S$, пусть $N_=(T)$ — число объектов множества A, которые обладают в точности свойствами из T (так что они не обладают никакими другими свойствами из $\overline{T} = S \setminus T$). Пусть $N_\geq(T)$ — число объектов множества A, обладающих по меньшей мере свойствами из T (и, возможно, какими-то другими свойствами).

Ясно, что тогда:

$$N_\geq(T) = \sum_{Y \supseteq T} N_=(Y), \tag{1.20}$$

$$N_=(T) = \sum_{Y \supseteq T} (-1)^{|Y\setminus T|} N_\geq(Y) \tag{1.21}$$

$$N_=(\varnothing) = \sum_{Y} (-1)^{|Y|} N_\geq(Y),$$

где Y пробегает все подмножества множества S.

В типичных приложениях принципа включений-исключений относительно легко вычислить $N_\geq(Y)$ для $Y \subseteq S$, так что нами получена окончательная формула для $N_=(T)$.

Распространенным частным случаем принципа включения-исключения является выполнение условия $N_=(T) = N_=(T^\circ)$, как только $|T| = |T^\circ|$. В рассматриваемом частном случае количество объектов, обладающих в точности заданным множеством свойств, зависит не от конкретного набора свойств, а только от числа рассматриваемых свойств. Таким образом, $N_\geq(T)$ зависит также только от $|T|$ и мы полагаем

$$a(n-i) = N_=(T) \tag{1.22}$$

и

$$b(n-i) = N_{\geq}(T), \tag{1.23}$$

если $|T| = i$.

Из формул (1.20) и (1.21) получаем, что формулы

$$b(m) = \sum_{k=0}^{m} \left(\begin{array}{c} m \\ k \end{array} \right) a(k), \;\; 0 \leq m \geq n \tag{1.24}$$

и

$$a(m) = \sum_{k=0}^{m} \left(\begin{array}{c} m \\ k \end{array} \right) (-1)^{m-k} b(k), \;\; 0 \leq m \leq n, \tag{1.25}$$

эквивалентны. Это еще одно отражение комбинаторных соотношений взаимности.

1.5.1 Задача о числе беспорядков (Задача о встречах)

В качестве канонического примера возможного приложения выведенной формулы включений-исключений рассмотрим знаменитую задачу о числе беспорядков или инверсий.

Пусть мы хотим найти число таких перестановок $\pi \in \sigma_n$, которые не имеют неподвижных точек, то есть ни один элемент не стоит на своем месте: $\pi(i) \neq i$ для всех $i \in \{1,\ 2,\ \ldots,\ n\}$. Таким образом, мы ищем число перестановок $a_1 a_2 \ldots a_n$ n чисел $\{1,\ 2,\ \ldots,\ n\}$, таких, что $a_i \neq i$, $i = 1,\ 2,\ \ldots,\ n$. Такие перестановки будем называть беспорядками. Обозначим их число $D(n)$. Тогда $D(0) = 1,\ D(1) = 0,\ D(2) = 1,\ D(3) = 2$.

Рассмотрим множество σ_n всех $n!$ перестановок $b_1 b_2 \ldots b_n$, и пусть свойство α_i перестановки π состоит в том, что $b_i = i$. Тогда $N(\alpha_{i_1},\ \ldots,\ \alpha_{i_s}) = (n-s)!$, $i_1 < \ldots < i_s$, поскольку они являются перестановками, в которых s объектов закреплены на своих позициях, а остальные объекты свободны. Число беспорядков есть N_0 и применение формулы включений-исключений дает равенство

$$N_0 = D(n) = n! - \left(\begin{array}{c} n \\ 1 \end{array} \right) (n-1)! + \left(\begin{array}{c} n \\ 2 \end{array} \right) (n-2)! + \ldots$$
$$\ldots + (-1)^k \left(\begin{array}{c} n \\ k \end{array} \right) (n-k)! + \ldots,$$

так как число способов выбора k неподвижных точек из n-элементного множества есть $\left(\begin{array}{c} n \\ k \end{array} \right)$.

Последнее выражение после упрощений можно переписать в виде:

$$N_0 = D(n) = n!(1 - 1 + \frac{1}{2!} - \frac{1}{3!} + \ldots + \frac{(-1)^n}{n!}). \tag{1.26}$$

Сумма в скобках в (1.26) есть начальный отрезок ряда $e^{-1} = \sum\limits_{i=0}^{\infty}(-1)^i\frac{1}{i!}$. Из формулы (1.26) видно, что $\frac{n!}{e}$ — хорошее приближение к $D(n)$, и, в действительности, нетрудно показать, что $D(n)$ — ближайшее целое к $\frac{n!}{e}$. Это означает, что беспорядки составляют фиксированную долю e^{-1} всех перестановок и, что удивительно, эта доля не зависит от n.

В задаче о беспорядках функция $b(i)$ (см. (1.23)) имеет очень специальное свойство $b(i) = i!$ — она зависит только от i, но не зависит от n. Эквивалентным образом, число перестановок, множество подвижных точек которых лежит во множестве T, зависит только от $|T|$, но не от n. Это означает, что формулу (1.26) можно переписать на языке конечных разностей (см. уравнение (1.6)) в виде

$$D(n) = \Delta^n x!|_{x=0}$$

(Сокращенная запись $D(n) = \Delta^n 0!$).

Так как число $b(i)$ перестановок, подвижные точки которых содержатся в некотором определенном i-элементном множестве, зависит только от i, то же верно и для числа $a(i)$ перестановок, имеющих в качестве множества подвижных точек некоторое определенное i-элементное множество. Из комбинаторных соображений ясно, что $a(i) = D(i)$.□

Из формулы (1.26) также немедленно следует, что при $n \geq 1$

$$D(n) = nD(n-1) + (-1)^n, \tag{1.27}$$

$$D(n) = nD(n-1) + D(n-2). \tag{1.28}$$

Значительного труда требует комбинаторное доказательство формулы (1.27), хотя прямое комбинаторное доказательство (1.28) совершенно просто. Приведем его.

Выберем и зафиксируем некоторый элемент $i > 1$ множества $\{1, 2, \ldots, n\}$. Все перестановки без неподвижных точек разобьем на два типа следующим образом. К первому типу K_1 отнесем перестановки, в которых i стоит на первом месте, но 1 не стоит на i-ом месте ($K_1 = \{\pi \in \sigma_n \mid \pi(i) = 1,\ \pi(1) \neq i\}$). Количество таких перестановок равно

$D(n-1)$. Ко второму типу K_2 отнесем перестановки, в которых i стоит на первом месте, а 1 на i-ом месте ($K_2 = \{\pi \in \sigma_n \mid \pi(i) = 1,\ \pi(1) = i\}$). Количество таких перестановок равно $D(n-2)$. Указанные типы перестановок не пересекаются, а их объединение по всем значениям i дает множество всех перестановок без неподвижных точек. Учитывая, что элемент i может быть выбран $n-1$ способом, получаем формулу (1.28).□

Рассмотрим пример, к которому непосредственно неприложимы предшествующие рассуждения, проведенные при решении задачи о числе беспорядков. Пусть $h(n)$ — число таких перестановок мультимножества $M_n = \{1^2,\ 2^2,\ \ldots,\ n^2\}$, никакие два последовательных члена которых не равны. Таким образом, $h(0) = 1,\ h(1) = 0,\ h(2) = 2$ (соответствующие перестановки есть 1212 и 2121). Пусть A - множество всех перестановок π мультимножества M_n и P_i для $1 \le i \le n$ — свойство перестановки π иметь последовательными членами два числа i. Тогда мы ищем $f_=(\varnothing) = h(n)$. Из соображений симметрии ясно, что при фиксированном n $f_\ge(T)$ зависит только от $i = |T|$, так что обозначим $g(i) = f_\ge(T)$. Ясно, что $g(i)$ равно числу перестановок π мультимножества $\{1,\ 2,\ \ldots,\ (i+1)^2,\ (i+2)^2,\ \ldots,\ n^2\}$ (замените каждое число $j \le i$, встречающееся в π, двумя последовательными членами, равными j), так что

$$g(i) = (2n-i)!2^{-(n-i)}$$

Заметим, что $b(i) = g(n-i) = (n+i)!2^{-i}$ не является функцией лишь от i, так что предшествующее рассуждение в действительности неприменимо. Однако из формулы (1.25) получаем, что

$$h(n) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) (-1)^{n-k}(n+k)!2^{-k} = \Delta^n (n+k)!2^{-k}\Big|_{k=0}$$

1.5.2 Количество сюръективных отображений

Вернемся теперь к задаче вычисления количества сюръективных отображений конечного множества X из n элементов на конечное множество Y из m элементов.

Теорема 1.14. *Число сюръективных отображений конечного множества X, $|X| = n$, на конечное множество Y, $|Y| = m$, то есть число функций $f : X \to Y$, таких, что $f(X) = Y$, равно*

$$f(n,m) = \sum_{k=0}^{m-1} (-1)^k \left(\begin{array}{c} m \\ k \end{array} \right) (m-k)^n.$$

Доказательство. Пусть $Y = \{y_1, \ldots, y_m\}$. Обозначим через α_i следующее свойство функции $f : X \to Y$: функция $f : X \to Y$, такова, что $y_i \notin f(x)$. Пусть F_{α_i} — множество функций, обладающих свойством α_i. Тогда очевидно, что

$$f(X) \neq Y \Leftrightarrow \bigcup_{i=1}^{m} F_{\alpha_i}$$

Число всех функций $f : X \to Y$ равно m^n.

Для произвольной последовательности $\alpha_{p_1}, \alpha_{p_2}, \ldots, \alpha_{p_i}$, такой что $1 \leq p_1 < \ldots < p_i \leq m$, $N(\alpha_{p_1}, \alpha_{p_2}, \ldots, \alpha_{p_i}) = (m-i)^n$. Именно столько имеется функций $f : X \to Y \setminus \{y_{p_1}, \ldots, y_{p_i}\}$. i-элементное подмножество $\{y_{p_1}, \ldots, y_{p_i}\}$ можно выбрать $\begin{pmatrix} m \\ i \end{pmatrix}$ способами, и, следовательно, по формуле включений-исключений имеем

$$f(n,m) = m^n + \sum_{i=1}^{m-1} (-1)^i \begin{pmatrix} m \\ i \end{pmatrix} (m-i)^n = \sum_{i=0}^{m-1} (-1)^i \begin{pmatrix} m \\ i \end{pmatrix} (m-i)^n$$

□

Следствие. На основании утверждения 1.12 и доказанной теоремы имеем

$$S(n,\ k) = \frac{1}{k!} f(n,\ k) = \frac{1}{k!} \sum_{i=0}^{k-1} (-1)^i C_k^i (k-i)^n$$

Таким образом, получено еще одно явное выражение для числа разбиений. Отметим, что разумно и иногда необходимо использовать это выражение в аналитических исследованиях свойств разбиений, хотя его эффективность в вычислениях самих чисел разбиений существенно ниже непосредственного использования рекуррентного соотношения (1.12).

1.5.3 Перестановки с ограничениями на местоположение

В задаче о числе беспорядков ищут число перестановок $\pi \in \sigma_n$, в которых для каждого i некоторые значения $\pi(i)$ запрещены (именно, $\pi(i) \neq i$). Рассмотрим более общую теорию таких перестановок. Традиционно ее описывают, используя шахматную терминологию.

Пусть $B \subseteq [n] \times [n]$. Множество B называют *доской*. Для $\pi \in \sigma_n$ определим *график* $G(\pi)$ перестановки π условием

$$G(\pi) = \{(i, \pi(i)),\ i \in [n]\}.$$

Определим теперь

1. $N_j = |\{\pi \in \sigma_n : j = |B \cap G(\pi)|\}|$,
2. $r_k =$ число k-подмножеств множества B, таких, что никакие два элемента не имеют общей координаты
3. $r_k =$ число способов разместить k не атакующих друг друга ладей на B

Мы можем отождествить перестановку $\pi \in \sigma_n$ с размещением n не атакующих друг друга ладей в квадратах $(i, \pi(i))$ доски $[n] \times [n]$. Тогда N_j есть число способов размещения n не атакующих друг друга ладей на доске $[n] \times [n]$, при которых в точности j из этих ладей находятся в B. Например, если $B = \{(1,1),\ (2,2),\ (3,3),\ (3,4),\ (4,4)\}$, то $N_0 = 6,\ N_1 = 9,\ N_2 = 7,\ N_3 = 1,\ N_4 = 1,\ r_0 = 1,\ r_1 = 5,\ r_2 = 8,\ r_3 = 5,\ r_4 = 1$. Наша цель — описать числа N_j и, в особенности N_0, в терминах чисел r_k. Определим многочлен N_n формулой

$$N_n(x) = \sum_j N_j x^j$$

Теорема 1.15. *Имеем*

$$N_n(x) = \sum_{k=0}^{n} r_k (n-k)! (x-1)^k \tag{1.29}$$

В частности,

$$N_0 = N_n(0) = \sum_{k=0}^{n} (-1)^k r_k (n-k)! \tag{1.30}$$

Доказательство. 1 Способ.

Пусть C_k есть число пар (π, C), где $\pi \in \sigma_n$ и C — k-подмножество множества $B \cap G(\pi)$. Для каждого j выберем N_j способами перестановку π так, чтобы $j = |B \cap G(\pi)|$, а затем C выберем $\binom{j}{k}$ способами. Следовательно, $C_k = \sum_j \binom{j}{k} N_j$.

С другой стороны, можно было бы сначала выбрать r_k способами множество C, а затем расширить его $(n-k)!$ способами до перестановки π. Следовательно, $C_k = r_k(n-k)!$. Поэтому

$$\sum_j \left(\begin{array}{c} j \\ k \end{array} \right) N_j = r_k(n-k)!,$$

или, эквивалентно,

$$\sum_j (y+1)^j N_j = \sum_k r_k(n-k)!y^k.$$

Полагая $y = x-1$, получаем желаемую формулу (1.29).

2 Способ.

Достаточно доказать формулу (1.29) в предположении, что x — положительное целое число. Левая часть формулы (1.29) подсчитывает число способов, которыми можно разместить не атакующие ладьи на доске и пометить каждую ладью на B элементами множества $[x]$. С другой стороны, такую конфигурацию можно получить, поместив k не атакующих ладей на участок B, пометив каждую из них элементом множества $\{2,\ 3,\ \ldots,\ x\}$, поместив $n-k$ добавочных ладей на доску $[n] \times [n](n-k)!$ способами и пометив новые ладьи на B единицами. Это устанавливает желаемое взаимно однозначное соответствие. □

Два доказательства теоремы 1.3 дают еще одну иллюстрацию двух комбинаторных способов доказательства равенства двух многочленов. Конечно, можно доказать формулу (1.30) прямым применением метода включений-исключений. Такое доказательство нельзя было бы рассматривать как комбинаторное, так как мы не построили явно взаимно однозначное соответствие между двумя множествами. Два доказательства, которые приведены выше, можно рассматривать как "полукомбинаторные так как они получаются из прямых формул взаимно однозначного соответствия, включающих параметры x и y соответственно, и затем мы получаем формулу (1.30), полагая $y = -1$ и $x = 0$.

1.16. Пример. (Снова задача о беспорядках). Возьмем $B = \{(1,1),\ (2,2),\ \ldots,\ (n,n)\}$. Мы хотим вычислить $N_0 = D(n)$. Ясно, что $r_k = \left(\begin{array}{c} n \\ k \end{array} \right)$,

так что

$$N_n(x) = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) (n-k)!(x-1)^k = \sum_{k=0}^{n} \frac{n!}{k!}(x-1)^k \Rightarrow$$

$$\Rightarrow N_0 = N_n(0) = \sum_{k=0}^{n} (-1)^k \frac{n!}{k!}$$

1.17. Пример. (Задача о супружеских парах (о гостях).) Эта известная задача формулируется следующим образом: сколькими способами можно рассадить за круглым столом n супружеских пар так, чтобы никакая пара не сидела рядом и женщины чередовались с мужчинами. Задача эквивалентна поиску числа перестановок $\pi \in \sigma_n$, для которых $\pi(i) \neq i$, $i + 1 (\mathrm{mod}\ n)$ для всех $i \in [n]$. Другими словами, мы ищем число N_0 для доски $B = \{(1,1),\ (2,2),\ \ldots,\ (n,n),\ (1,2),\ (2,3),\ \ldots,\ (n-1,n),\ (n,1)\}$. Взглянув на рисунок доски B, мы видим, что r_k равно числу способов, которыми можно выбрать k из $2n$ расположенных на окружности точек так, чтобы среди них не было двух последовательных (соседних).

Лемма 1.18. *Число способов, которыми можно выбрать k точек из m точек на окружности так, чтобы среди них не было двух последовательных (соседних) точек, равно*

$$\frac{m}{m-k} \left(\begin{array}{c} m-k \\ k \end{array} \right).$$

Доказательство. 1 способ

Пусть $f(m,\ k)$ — искомое число, и пусть $g(m,\ k)$ — число способов выбрать k не последовательных точек из m точек, расположенных по окружности, а затем покрасить k выбранных точек в красный цвет и одну из неокрашенных точек покрасить в синий цвет. Ясно, что

$$g(m,\ k) = (m-k)f(m,\ k).$$

Но можно вычислить $g(m,\ k)$ также следующим образом. Сначала покрасим одну точку в синий цвет m способами. Теперь нам нужно покрасить в красный цвет k точек, выбранных из линейного массива $m-1$ точек так, чтобы среди них не было двух последовательных. Одним из способов дальнейших рассуждений может быть следующий. Расположим $m-1-k$ неокрашенных точек в линию и вставим k красных точек в $m-k$ промежутков между неокрашенными точками (считая начало и конец).

Это можно сделать $\begin{pmatrix} m-k \\ k \end{pmatrix}$ способами. Следовательно, $g(m,\ k) = m \begin{pmatrix} m-k \\ k \end{pmatrix}$. Откуда

$$f(m,\ k) = \frac{m}{m-k} \begin{pmatrix} m-k \\ k \end{pmatrix}.$$

Предложенное доказательство основано на некотором общем принципе перехода от кругового к линейному массиву. Этот принцип широко используется при решении комбинаторных задач.

2 способ

Пометим точки числами $1,\ 2,\ \ldots,\ m$ в возрастающем по часовой стрелке порядке. Мы хотим покрасить k из них в красный цвет, так чтобы не было двух последовательных красных точек. Сначала подсчитаем число возможностей, при которых точка 1 не окрашена в красный цвет. Расположим $m-k$ неокрашенных точек по кругу, пометив одну из них единицей и вставим k красных точек в $m-k$ промежутков между неокрашенными точками $\begin{pmatrix} m-k \\ k \end{pmatrix}$ способами. С другой стороны, если точка 1 будет покрашена в красный цвет, то расположим $m-k+1$ точек по кругу, покрасим одну из этих точек в красный цвет и пометим ее 1, а затем вставим $\begin{pmatrix} m-k-1 \\ k-1 \end{pmatrix}$ способами $k-1$ красную точку на $m-k-1$ разрешенных мест.

Следовательно,

$$f(m,\ k) = \begin{pmatrix} m-k \\ k \end{pmatrix} + \begin{pmatrix} m-k-1 \\ k-1 \end{pmatrix} = \frac{m}{m-k} \begin{pmatrix} m-k \\ k \end{pmatrix}$$

□

На основании доказанной леммы имеем

Утверждение 1.19. *Многочлен $N_n(x)$ для доски*

$$B = \{(i,i), (i,\ i+1(\mathrm{mod}\ n)) : 1 \le i \le n\}$$

дается формулой

$$N_n(x) = \sum_{k=0}^{n} \frac{2n}{2n-k} \begin{pmatrix} 2n-k \\ k \end{pmatrix} (n-k)!(x-1)^k.$$

В частности, число N_0 перестановок $\pi \in \sigma_n$, для которых $\pi(i) \neq i$, $i+1 \pmod n$ при $1 \leq i \leq n$, дается формулой

$$N_0 = \sum_{k=0}^{n} \frac{2n}{2n-k} \left(\begin{array}{c} 2n-k \\ k \end{array} \right) (n-k)!(-1)^k$$

1.6 Системы представителей множеств

Рассмотрим один из комбинаторных подходов к характеристике структуры конечных множеств. Уже по названию можно понять, что основной идеей является замена системы множеств собранием их представителей. Мы рассмотрим здесь несколько теорем, которые гарантируют существование определенных выборов при соответствующих предположениях. Они интересны сами по себе и могут быть использованы в качестве теорем существования в различных комбинаторных задачах.

1.6.1 Системы различных представителей

Пусть S — конечное множество из m элементов, $|S| = m$. $P(S)$ — множество всех его подмножеств.

Определение. Пусть $M(S) = (S_1,\ S_2,\ \ldots,\ S_n)$ некоторая совокупность подмножеств из $P(S)$, необязательно различных, $a = (a_1,\ \ldots,\ a_n)$ — последовательность элементов из S, такая, что все элементы a_i, $i = 1,\ 2,\ \ldots,\ n$ различны: $a_i \neq a_j$, $i \neq j$. Если при этом $a_i \in S_i$, то говорят, что элемент a_i *представляет множество* S_i, а вся совокупность $(a_1,\ \ldots,\ a_n)$ называется *системой различных представителей* (с.р.п.) для $M(S)$. □

Заметим еще раз, что в с.р.п. если $i \neq j$, то $a_i \neq a_j$, если даже $S_i = S_j$. Если множество появляется несколько раз, то всякий раз оно должно иметь представителя, отличного от всех других.

Сразу же оказывается, что с.р.п. может существовать не для всех совокупностей множеств. Если в конечной системе множества непусты и различны, то с.р.п., очевидно, существует. Возьмем более сложный случай:

$$S = \{a,\ b,\ c,\ d,\ f\}$$

$$M(S):\ S_1 = \{a,\ b,\ c,\ d\},\ S_2 = \{a,\ b,\ f\},\ S_3 = S_4 = \{b,\ f\}.$$

В этом случае существует две с.р.п.: $(c,\ a,\ b,\ f)$, $(d,\ a,\ f,\ b)$. Но стоит изменить лишь одно из подмножеств, например, вместо S_2 взять $S_2 = (d,\ f)$ и мы уже не сможем получить ни одной с.р.п.

Теорема 1.20 (Теорема о различных представителях). *Подмножества* $S_1,\ S_2,\ \ldots,\ S_n$ *имеют с.р.п. тогда и только тогда, когда удовлетворяется следующее* условие С*: среди элементов любого конечного числа* k *множеств* S_i *имеется по меньшей мере* k *различных элементов; иными словами множество* $S_{i_1} \cup S_{i_2} \cup \ldots \cup S_{i_k}$ *состоит не менее чем из* k *элементов для всех* $k = 1,\ 2,\ \ldots,\ n$.

Замечание. Если общее число множеств конечно, а сами множества бесконечны, то теорема остается в силе, так как можно, очевидно, отбросить все элементы, кроме n, в каждом S_i, не нарушая условия С.

Доказательство. Практически очень трудно проверить, выполняется ли в данном конкретном случае условие С. Доказательство, основанное на полной математической индукции, не дает ничего, что помогло бы найти с.р.п. Это не удивительно, теоремы существования чаще всего появляются там, где бывает трудно или невозможно найти алгоритм, приводящий к нахождению решения.

Очевидно, что условие С есть *необходимое* условие, потому что если различные представители существуют, то для любых k множеств представителями будут k различных элементов, содержащихся в этих множествах.

В качестве доказательства *достаточности* приведем алгоритм, который позволяет построить с.р.п. или показать, что такой системы для данного набора множеств не существует.

Пусть задано n множеств и выполнено условие С. Требуется найти для них с.р.п. или показать, что этой системы не существует, если условие С не выполняется.

Пронумеруем множества $S_1,\ \ldots,\ S_n$ и зафиксируем порядок, в котором они занумерованы. Выберем произвольный элемент a_1 из S_1 в качестве его представителя. Поочередно будем выбирать представителей других множеств: $a_2 \in S_2,\ a_3 \in S_3,\ \ldots$, заботясь лишь о том, чтобы каждый из них был отличен от любого другого, выбранного в качестве представителя, элемента. Если этот процесс удастся довести до S_n, то мы получим с.р.п.

Мы можем, однако, достичь множества S_r, все элементы которого $b_1,\ b_2,\ \ldots,\ b_t$ были уже использованы как представители множеств $S_1,\ S_2,\ \ldots,\ S_{r-1}$. Это, однако, еще не означает, что с.р.п. не существует. Тогда построим последовательность вспомогательных множеств $T_0,\ T_1,\ \ldots$.

Зафиксируем порядок нумерации элементов множества S_r:

$$S_r = \{b_1,\ b_2,\ \ldots,\ b_t\} \subset \{a_1,\ a_2,\ \ldots,\ a_{r-1}\}.$$

Определим множество T_0 состоящим из элементов множества S_r с фиксированным выше порядком нумерации элементов:

$$T_0 = \{b_1,\ b_2,\ \ldots,\ b_t\}.$$

Будем двигаться по списку элементов множества T_0 и последовательно строить вспомогательные множества $T_1,\ T_2,\ \ldots$, до тех пор, пока не обнаружим элемент, не использованный в качестве представителя или не исчерпаем все множества. Обозначим символом $S(b_i)$ множество, представителем которого является элемент b_i.

Пусть теперь множество T_1 состоит из элементов множества, представителем которого является b_1, за исключением самого элемента b_1 и всех остальных использованных в качестве представителей элементов:

$$T_1 = S(b_1) \setminus T_0.$$

Множество T_1 может быть и пустым, но если оно не пусто, то модифицируем множество T_0, приписав к списку его элементов непосредственно за $b_1,\ \ldots,\ b_t$ элементы множества T_1, обозначенные как $b_{t+1},\ \ldots,\ b_s$:

$$T_0 = T_0 \times T_1 = \{\underbrace{b_1,\ \ldots,\ b_t}_{T_0}, \underbrace{b_{t+1},\ \ldots,\ b_s}_{T_1}\}$$

Далее, если i-ый элемент b_i есть представитель множества $S_j = S(b_i)$, то мы построим множество T_i, состоящее из тех элементов S_j, которые еще не использованы, и запишем эти элементы после элементов, уже использованных:

$$T_i = S(b_i) \setminus T_0; \quad T_0 = T_0 \cdot T_i.$$

Так мы сможем поступать до тех пор, пока либо:

1. мы достигли некоторого элемента $b_i \in S_j$ $(i > t,\ j < r)$, либо
2. последовательность T_0 исчерпывается элементами $b_1,\ \ldots,\ b_s$ как представителями множеств.

Во втором случае мы должны быть убеждены, что с.р.п. не существует. В самом деле, получена некоторая последовательность элементов $b_1,\ \ldots,\ b_v$, исчерпывающая множество всех различных представителей. Эти элементы являются представителями v множеств $(v < n)$. По построению каждый элемент этих v множеств содержится в последовательности. Но тогда эти

множества (v штук), а также множество S_r образуют $v+1$ множество, которые содержат только v различных элементов, таким образом мы нашли множества, нарушающие условие С.

Если же имеет место случай 1, мы на некотором этапе находим элемент $b_i = b_{i_1}$ $(i_1 > t)$, который входит во множество S_{j_1}, представителем которого является выбранный ранее другой элемент b_{i_2}, причем $i_2 < i_1$. Если $i_2 > t$, то значит $b_{i_2} \in S_{j_2}$, а представителем множества S_{j_2} является $b_{i_3} \in S_{j_3}$ $(i_3 < i_2)$ и так далее. Таким образом, возникает последовательность b_{i_1}, b_{i_2}, $\ldots$, b_{i_m}, индексы которой убывают $(i_m \leq t)$, причем в этой последовательности каждый ее член входит во множество, представителем которого является следующий член. Но теперь мы можем заменить представителей: возьмем b_{i_1} в качестве представителя S_{j_1}, b_{i_2} — в качестве представителя S_{j_2}, $\ldots$, $b_{i_{m-1}}$ — для $S_{j_{m-1}}$. Элемент b_{i_m} в результате такой замены освобождается для выбора в качестве представителя S_r. Итак, мы, действуя тем же путем, найдем представителя S_{r+1} и так далее. Наши построения закончатся либо выбором системы различных представителей, либо мы обнаружим систему множеств, для которой нарушается условие С. □

1.21. Пример. Пусть имеется следующая система множеств:
$S_1 = \{1,2\}$; $S_2 = \{2,3\}$; $S_3 = \{3,4\}$; $S_4 = \{5,2\}$; $S_5 = \{4,6\}$; $S_6 = \{1,5\}$.

Будем обозначать выбор элемента b в качестве представителя множества S записью $b = P(S)$.

Последовательно выберем следующих представителей множеств:

$$1 = P(S_1);\ \ 2 = P(S_2);\ \ 3 = P(S_3);\ \ 5 = P(S_4);\ \ 4 = P(S_5).$$

Дойдя таким образом до множества S_6, мы обнаруживаем, что все элементы множества S_6 уже использованы как представители других множеств.

Строим последовательность множеств $T_0, T_1, \ldots$:

$$\begin{array}{ll} T_0 = \{1,\ 5\} & \\ T_1 = S(1) \setminus T_0 = \{2\} & T_0 = \{1,\ 5,\ 2\} \\ T_2 = S(5) \setminus T_0 = \varnothing & \\ T_3 = S(2) \setminus T_0 = \{3\} & T_0 = \{1,\ 5,\ 2,\ 3\} \\ T_4 = S(3) \setminus T_0 = \{4\} & T_0 = \{1,\ 5,\ 2,\ 3,\ 4\} \\ T_5 = S(4) \setminus T_0 = \{6\} & T_0 = \{1,\ 5,\ 2,\ 3,\ 4,\ 6\} \end{array}$$

Таким образом, передвигаясь по элементам множества

$$T_0 = \{\underbrace{1,5}_{S_6},\ \underbrace{2}_{S_1},\ \underbrace{3}_{S_2},\ \underbrace{4}_{S_3},\ \underbrace{6}_{S_5}\}$$

мы, наконец, достигаем элемента 6, который не является представителем никакого множества. Мы нашли последовательность элементов 1, 5, 2, 3, 4, 6, такую что:

- элемент $6 \in S_5$, а $P(S_5) = 4$;
- элемент 4 вошел в последовательность как элемент множества S_3, $4 \in S_3$, а $P(S_3) = 3$;
- элемент 3 вошел в последовательность как элемент множества S_2, $3 \in S_2$, а $P(S_2) = 2$;
- элемент 2 вошел в последовательность как элемент множества S_1, $2 \in S_1$, а $P(S_1) = 1$.

Теперь мы можем заменить представителей, положив

$$6 = P(S_5);\ \ 4 = P(S_3);\ \ 3 = P(S_2);\ \ 2 = P(S_1)$$

освободившийся элемент 1 может быть использован в качестве представителя множества S_6.

1.6.2 Системы общих представителей

Идея замены множеств их представителями оказалась плодотворной и получила последующее развитие. Системы представителей выделяются с учетом условий задач или целей теоретических обобщений. Например, задачи о разбиении множеств привели к понятию систем общих (одновременных) представителей.

Пусть даны два различных разбиения одного и того же множества S на n непересекающихся непустых подмножеств:

$$S = A_1 \cup \ldots \cup A_n = B_1 \cup \ldots \cup B_n;$$

$$A_i \cap A_j = \varnothing,\ i \neq j,\ i, j \in [n];$$

$$B_i \cap B_j = \varnothing,\ i \neq j,\ i, j \in [n].$$

Определение. Если существует подмножество $O \subseteq S$, состоящее из n различных элементов $x_1, \ldots, x_n$, которые являются одновременными представителями множеств A_i и B_j то оно называется *системой общих представителей (с.о.п.)*.

Теорема 1.22. *Два разбиения множества* S, $S = A_1 \cup A_2 \cup \ldots \cup A_n$ *и* $S = B_1 \cup \ldots \cup B_n$ *тогда и только тогда имеют с.о.п., когда любые* m *из множеств* B_i *содержатся не менее, чем в* m *из множеств* A_j, $m \leq n$.

Доказательство. Необходимость, как и в случае с.р.п., очевидна. Достаточность доказывается простым сведением к теореме о с.р.п..

Рассмотрим $A_1, \ldots, A_n \subseteq S$.

Для каждого B_i, $i = 1, 2, \ldots, m$ определим множество S_i, состоящее из A_j $(j = 1, \ldots, m)$ с такими индексами j, что $A_j \cap B_i$ не пусто:

$$S_i = \{\cup A_j \mid A_j \cap B_i \neq \varnothing\}.$$

Получим $M(S) = \{S_1, S_2, \ldots, S_n\}$.

Для $M(S)$ существует с.р.п. .

Выбор различных представителей для каждого B_i дает свое A_j, причем пересечение между ними не пусто. В этом пересечении можно выбрать хотя бы один элемент, общий для A_j и B_i, т.е. общего представителя. $\square$

Глава 2

Функции алгебры логики

Согласно одному из самых распространенных определений, логика есть анализ методов рассуждений. Изучая эти методы, логика прежде всего интересуется формой доводов, а не их содержанием в том или ином рассуждении. Практические приложения методов математической логики к проектированию и эксплуатации вычислительных и управляющих систем хорошо известны.

«Вытекает ли истинность заключения из истинности посылок?» — таков основной вопрос математической логики. Необходимо исследовать язык логики и математики, чтобы разобраться в том, какие в ней могут быть употреблены символы, как из этих символов составляются утверждения и доказательства, что может и что не может быть доказано, если исходить из тех или иных аксиом и правил вывода. Содержанием математической логики является изучение языка математики. Разумеется, забота о языке и постоянная его перестройка для приведения в соответствие с меняющимся состоянием знаний характерна для любой естественной науки (достаточно вспомнить «флогистон» и «мировой эфир» в физике). Тем не менее, то пристальное критическое рассмотрение, которому математика подвергла свои средства выражения и самое себя, представляется уникальным.

Причина этого заключается, конечно, в том, что все остальные науки имеют предметом изучения внешний по отношению к ним реальный мир, и эволюция языка науки определяется постоянным сравнением научного описания с описываемой реальностью. Попытка применить эту же схему к математике сразу наталкивается на принципиальные трудности: в каком смысле числа и множества реальны? Столь же неясным при внимательном

рассмотрении становится ответ на вопрос, что есть истинность математического рассуждения. Положение дел здесь можно сравнить с понятием «элементарной частицы» в физике. Элементарные частицы — это не те частицы, которые являются последними кирпичиками анализа,— скорее это те объекты, дальнейший анализ которых обнаруживает существенное повышение уровня сложности вместо его понижения. «Элементарность» понятия истины имеет сходные черты.

Фундаментальные результаты Гёделя и других авторов называются теоремами о неполноте, неразрешимости, независимости. Такого типа результаты, представляющие очевидный общегуманитарный интерес, не имели прецедентов в развитии философской мысли до XX века, и являются существенным вкладом естественных наук в фонд гуманитарных, подчеркивая невозможность полной формализации математики и обнажая ее глубокий гуманитарный смысл.

Физическое рассуждение правильно, если полученные с его помощью выводы совпадают с реально наблюдаемыми фактами. Критерием истинности математического рассуждения является лишь его логическая безукоризненность, выполнение на всех этапах рассуждения устанавливаемых самим математиком правил вывода, относящихся к вполне определенной ветви математической науки — математической логике. При этом на сегодняшний день мы имеем вовсе не один-единственный набор правил вывода, а много разных таких наборов (аналогично существованию геометрии Евклида, геометрии Лобачевского, геометрии Римана в логике существуют такие подходы к формализации как конструктивизм, интуиционизм). Своеобразная природа математики, управляемой законами ею же самой себе указываемыми, характеризует особенность этой ветви человеческой культуры, но не ее единственность. Искусство также само себе диктует правила игры: разработанные Леонардо да Винчи и Дюрером правила перспективы, каноны построения византийских или русских икон, правила «икебана». Эти каноны не менее непреложны и жестки, чем аксиомы геометрии, и изменить эти правила может лишь выдающийся художник, подобно тому как новые области «математической вселенной» нам открывали Ньютон, Гильберт, Лобачевский, Брауэр.

Одним из основных проявлений происходящего в наши дни общенаучного переворота, связанного с внедрением ЭВМ и получившего у журналистов кодовое наименование «научно-техническая революция», является колоссальная математическая экспансия, вторжение математики во все новые, ранее ею никак неконтролируемые территории. Математическими методами широко пользуются представители самых разных — как есте-

ственнонаучных, так и гуманитарных областей знания: биологи и филологи, экономисты и юристы. Все это сделало понимание путей использования математического аппарата во внематематических исследованиях чуть ли не одним из важнейших элементов общей культуры, а владение терминами «математическая структура» и «математическая модель» — необходимыми атрибутами образованного человека.

Предметом логики не является внешний мир, но лишь системы его осмысления. Логика одной из таких систем — математики — в силу своей нормализованности представляет подобие трафарета, который можно накладывать на любую другую систему. Соответствие или расхождение этого трафарета с системой, однако, не служит критерием ее пригодности либо мерилом ценности.

Еще Лейбниц (1789-1857) высказал идею формализации всей математики на основе создания «логического исчисления» — своеобразного математического языка. Записав все исходные допущения на таком языке специальными символами, похожими на математические, он надеялся заменить рассуждения вычислениями. В тридцатых годах нашего века Давид Гильберт выступил с программой обоснования математики на базе финитных методов математической логики. Окончательно эти надежды развеяли упомянутые выше результаты Гёделя (1937) о неполноте исчисления предикатов и формальной арифметики.

Мы будем изучать одну из простейших моделей математической логики — алгебру высказываний или алгебру логики.

Первыми математическими работами, заложившими основу современной алгебры логики, были работы Джорджа Буля (1815-1864) и Аугстаса де Моргана (1806-1873). Названия этих работ несомненно заслуживают упоминания.

Джордж Буль:
«Математический анализ логики, являющийся очерком, касающимся исчисления дедуктивных рассуждений», (1847 г.),
«Исследования законов мысли. на которых основываются математические теории логики и вероятностей», (1854 г.).

Аугустус де Морган:
«Формальная логика или исчисление выводов, необходимых и возможных» (1847 г.).

2.1 Элементарные высказывания

В логике под «высказыванием» понимают то, что выражается, как говорят в лингвистике, посредством осмысленного утвердительного предложения, то есть повествовательного предложения, о котором можно (по крайней мере в пределах определенного контекста) говорить, что оно истинно или ложно.

Примеры элементарных высказываний:
Снег белый.
Париж — столица Италии.
Все люди смертны.
Сократ — человек.
Если снег горит, то остается зола.
Если на улице идет дождь, то влажность выше, чем при солнечной сухой погоде.

Не всякое предложение является высказыванием. Так, к высказываниям не относятся вопросительные и восклицательные предложения, поскольку говорить об их истинности или ложности нет смысла. Предложения «Шел снег», «Площадь комнаты равна 20 м^2», «$a^2 = 4$» не являются высказываниями; для того, чтобы имело смысл говорить об их истинности или ложности, нужны дополнительные сведения: когда и где шел снег, о какой конкретной комнате идет речь, какое число обозначено буквой a. В последнем примере a может не обозначать конкретного числа, а быть переменной, вместо которой можно подставлять элементы некоторого множества, называемые значениями переменной.

Из двух данных предложений можно образовать новое предложение с помощью союзов «и», «или», «если ... то», «тогда и только тогда, когда». С помощью частицы «не» или словосочетания «неверно, что» из одного данного предложения можно получить новое. Союзы «и», «или», «если ... то», «тогда и только тогда, когда» и частицу «не» называют логическими связками.

В высказывании нас прежде всего интересует его истинностное значение, то есть является оно истинным или ложным. Чтобы ответить на этот вопрос, нам ничего не надо знать о составляющих высказываниях, кроме их истинностных значений. Эта информация полностью определяет истинностное значение сложного высказывания.

Для обозначения лжи мы будем использовать символ 0, а для обозначения истины — символ 1.

Элементарные высказывания, таким образом, мы будем обозначать сим-

волами переменных, принимающих значения 0 или 1. Такие переменные будем называть логическими или булевыми переменными.

Пусть $U = \{u_1,\ u_2,\ \ldots,\ u_n,\ \ldots\}$ — исходный алфавит переменных. 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 и другие символы с индексами. Итак, логические переменные (элементарные высказывания) имеют вид:

$x,\ y,\ z,\ \ldots$;
$x_1,\ x_2,\ x_3,\ \ldots$;
$x \in \{\text{истина, ложь}\} = \{1,\ 0\}$;
истина $= 1$;
ложь $= 0$.

2.2 Элементарные логические операции(функции)

Определение. Функция $f(\tilde{x}_n)$, определенная на множестве

$$B^n = \{\tilde{x}_n | \tilde{x}_n = (x_1,\ \ldots,\ x_n),\ x_i \in \{0,1\}$$

и принимающая значения из множества $\{0,\ 1\}$, называется *функцией алгебры логики* или *булевой функцией*.

Очевидно, что для задания функции $f(\tilde{x}_n)$ достаточно указать, какое значение принимает функция при каждом из наборов значений аргументов, то есть выписать таблицу:

$x_1,\ \ x_2,\ \ \ldots,\ \ x_{n-1},\ \ x_n$	$f(x_1,\ x_2,\ \ldots,\ x_{n-1},\ x_n)$
$0\,0\ldots0\,0$	$f(0,\ 0,\ldots0,\ 0)$
$0\,0\ldots0\,1$	$f(0,\ 0,\ldots0,\ 1)$
$0\,0\ldots1\,1$	$f(0,\ 0,\ldots1,\ 1)$
$\ldots$	$\ldots$
$1\,1\ldots1\,1$	$f(1,\ 1,\ldots,1,\ 1)$

Таблица 2.1

Легко видеть, что n переменных в совокупности принимают 2^n различных значений. Для удобства мы употребляем стандартное расположение наборов значений переменных: если набор рассматривать как двоичную

запись числа, то их расположение соответствует естественному расположению чисел $0,\ 1,\ \ldots,\ 2n-1$. Каждая функция $f(x_1,\ x_2,\ \ldots,\ x_n)$ определяет отображение $B^2 \times B^2 \ldots \times B^2 \to B^2$. Поэтому естественно интерпретировать f как символ, обозначающий это отображение, а $x_1,\ x_2,\ \ldots,\ x_n$ как названия столбцов. В этом случае функции $f(x_1,\ x_2,\ \ldots,\ x_n)$ и $f(y_1,\ y_2,\ \ldots,\ y_n)$ будут задавать одно и то же отображение, а их таблицы будут отличаться только, быть может, названиями столбцов. Обозначим через P_2 множество всех функций алгебры логики над алфавитом U, содержащее также и константы 0 и 1. Если зафиксировать n переменных $x_1,\ x_2,\ \ldots,\ x_n$, то различные таблицы будут отличаться лишь значениями в правом столбце. Поэтому справедливо следующее утверждение.

Теорема 2.1. *Число $p_2(n)$ всех функций из P_2, зависящих от переменных $x_1,\ x_2,\ \ldots,\ x_n$, равно 2^{2^n}.*

Здесь следует обратить внимание на одно обстоятельство. 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 перебор функций из данного множества. Однако числа $p_2(n)$ с ростом n быстро растут: $p_2(1) = 4,\quad p_2(2) = 16,\quad p_2(3) = 256,\quad p_2(4) = 65536,\quad \ldots$ Таким образом, уже при сравнительно небольших значениях n $(n \geq 6)$ перебор становится практически неосуществимым даже с использованием вычислительной техники. С ростом числа аргументов n таблица, задающая функцию, сильно усложняется. Так при $n = 64$ для заполнения такой таблицы со скоростью 10^9 строк в секунду понадобится около 300 лет. Поэтому несмотря на простоту табличного задания функций алгебры логики, такой инструмент исследования функций крайне неэффективен, если не бесполезен, при больших значениях n. Необходимость разработки аналитического задания и методов исследования функций алгебры логики очевидна.

Определение. Переменная $x_i (1 \leq i \leq n)$ функции $f(x_1,\ x_2,\ \ldots,\ x_{i-1},\ x_i,\ x_{i+1},\ \ldots,\ x_n)$ из P_2 называется *существенной*, если можно указать такие наборы

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

и

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

значений переменных, что $f(\tilde{\alpha}) \neq f(\tilde{\beta})$. В противном случае переменную x_i называют *несущественной* или *фиктивной* переменной функции $f(x_1,\ x_2,\ \ldots,\ x_{i-1},\ x_i,\ x_{i+1},\ \ldots,\ x_n)$.

Определение. Две функции $f(\tilde{x}_n)$ и $g(\tilde{y}_n)$ называются *равными*, если множества их существенных переменных совпадают и на любых двух наборах $\tilde{\alpha}_n$ и $\tilde{\beta}_n$, различающихся быть может только значениями несущественных переменных, значения функций одинаковы: $f(\tilde{\alpha}_n) = g(\tilde{\beta}_n)$. Если $f(\tilde{x}_n)$ и $g(\tilde{y}_n)$ — равные функции, то одну из них можно получить из другой путем добавления и/или изъятия несущественных переменных.

Построение аналитической теории функций алгебры логики начнем с элементарных функций (операций).

Данные операции (функции) часто употребляются в математической логике и кибернетике и играют такую же роль, как, например, $x + y$ в арифметике, x^2 в алгебре, $\sin x$ и $\exp x$ в анализе, поэтому их можно считать элементарными функциями.

Определим логические или булевские операции (функции) над элементарными высказываниями или логическими (булевскими) переменными.

1. Константы (нульместные операции):
   0 — тождественный нуль (тождественная ложь),
   1 — тождественная единица (тождественная истина).

2. Операции над одной переменной (одноместные, унарные операции):
   отрицание, обозначается как $\bar{x}$ или $\neg x$, читается «не x» или «отрицание x». Эту операцию можно задать следующей таблицей, в которой указано, какое значение функции (операции) соответствует каждому из значений переменной x:

x	¬ x
0	1
1	0

3. Операции над двумя переменными (двухместные, бинарные операции)

x	y	$x \wedge y$	$x \vee y$	$x \to y$	$x \equiv y$	$x + y$	$x \mid y$	$x \downarrow y$
0	0	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	0

Приведем названия и другие обозначения перечисленных в таблице функций.

Определение. Функция $\mathbf{x \wedge y}$ называется *конъюнкцией* x и y, обозначается $x \wedge y$, или $x\&y$, или $x \cdot y$, или xy, или $\min(x, y)$ и часто читается как «x и y».

Определение. Функция $\mathbf{x \vee y}$ называется *дизъюнкцией* x и y, обозначается $x \vee y$ или $\max(x, y)$ и часто читается как «x или y».

Определение. Функция $\mathbf{x \to y}$ называется импликацией x и y, обозначается $x \to y$ или $x \supset y$, часто читается как «x имплицирует y» или «из x следует y».

Определение. Функция $\mathbf{x \equiv y}$ называется эквивалентностью (или эквиваленцией) x и y, обозначается $x \equiv y$, или $x\ y$, или $x \leftrightarrow y$ и часто читается «x эквивалентно y».

Определение. Функция $\mathbf{x + y}$ называется суммой по модулю 2 (или булевой суммой) x и y, обозначается $x + y$, или $x \oplus y$ и часто читается «x плюс y».

Определение. Функция $\mathbf{x|y}$ называется штрихом (Шеффера) x и y, обозначается $x|y$ и часто читается «x штрих y», «не x или не y». В технической литературе функция $\mathbf{x|y}$ называется обычно «не — и» или антиконъюнкцией, так как она равна отрицанию конъюнкции.

Определение. Функция $\mathbf{x \downarrow y}$ называется стрелкой (Пирса) x и y, обозначается $x \downarrow y$ и часто читается «x стрелка y», «ни x, ни y» или «не x и не y». В технической литературе функция $\mathbf{x \downarrow y}$ называется обычно «не — или» или антидизъюнкцией, так как она равна отрицанию дизъюнкции.

Символы операций, участвующие в обозначениях элементарных функций, называются логическими связками (или просто связками).

Имея запас элементарных функций. мы можем строить из них более сложные с помощью введенных логических связок. Например,

$$((x \to y)|(y \to z)) \vee (\bar{y} \cdot \bar{z}) = f(x,\ y,\ z).$$

Очевидно, что не всякая последовательность символов переменных, знаков логических операций и скобок будет формулой алгебры логики. Так, например, последовательности $\to\ \vee\ x + y\ x|y \equiv z$ не могут рассматриваться как правильные формулы алгебры логики. Дадим индуктивное определение формулы.

Определение. Пусть U — множество переменных. Тогда множество формул алгебры логики над U определяется следующим образом:

1. Всякая переменная — формула.

2. Константы 0 и 1 — формулы.

3. Если A — формула, то $\rceil A$ (или в другой записи $\bar{A}$) — формула.

4. Если A и B — формулы, то $(A \wedge B)$, $(A \vee B)$, $(A \to B)$, $(A + B)$, $(A \equiv B)$, $(A|B)$, $(A \downarrow B)$ — формулы.

5. Формулами являются те и только те выражения, которые могут быть получены из констант, переменных и логических связок за конечное число шагов 1 — 4.

В пунктах 1 и 2 определены элементарные формулы, в пунктах 3 и 4 заданы правила образования из любых данных формул новых. Определения такого типа называются индуктивными определениями. Во всяком индуктивном определении имеются прямые пункты (п.п. 1—4) и косвенный пункт. В прямых пунктах задаются объекты, которые в дальнейшем именуются определяемым термином (в данном случае формулами алгебры высказываний). В косвенном пункте говорится, что такие объекты исчерпываются заданными в прямых пунктах. Среди прямых пунктов имеются базисные пункты (п. 1 и п. 2) и индуктивные пункты (в данном случае п. 3 и п. 4). В базисных пунктах прямо указываются объекты, которые в дальнейшем именуются определяемым термином. В индуктивных пунктах даются правила построения из любых объектов, определенных в базисных пунктах, новых объектов, которые также будут именоваться этим термином.

Придавая всевозможные значения всем входящим в формулу алгебры логики переменным, мы получим таблицу значений функции. В этом смысле мы будем говорить, что *заданная формула реализует функцию алгебры логики.*

Две формулы A и B (или f_1 и f_2) будем называть равносильными (равными) и писать $A = B$ (или $f_1 = f_2$), если они реализуют одну и ту же функцию алгебры логики.

Очевидно, что если A' — подформула формулы A, то при замене любого ее вхождения на равную формулу B формула A перейдет в формулу B, которая будет равна A.

Обычно принимаются следующие соглашения для сокращения записи формул:

- внешние скобки у формул опускаются;
- формула $\rceil$ записывается как $\bar{A}$;
- считается, что операция отрицания старше любой другой операции, то есть если за связкой $\rceil$ следует символ переменной (буква), то отрицание относится только к этой переменной, если же сразу после

отрицания $\neg$ открывается скобка, то отрицание относится ко всему заключенному в скобки выражению;

- формула $A \wedge B$ записывается как $A \cdot B$, или $A\&B$, или AB;
- связка $\wedge$ считается старше (сильнее) любой другой двухместной связки.

Эти соглашения позволяют, например, записать формулу

$$((((\neg x) + y) \wedge z) \to ((x \wedge (\neg y)) \vee z))$$

в виде

$$(\bar{x} + y) \to (x\bar{y} \vee z).$$

2.3 Алгебраические свойства элементарных операций

Важнейшими алгебраическими свойствами логических операций являются, как обычно, такие свойства, как коммутативность, ассоциативность и дистрибутивность одних операций относительно других.

1. **Коммутативность** (или перестановочность) операции * означает, что $x * y = y * x$. Логическая операция * коммутативна, если связка * принадлежит следующему множеству связок (существенно только, чтобы символ * в равенстве всюду имел один и тот же смысл):

$$* \in \{\vee,\ \wedge,\ +,\ \equiv,\ |,\ \downarrow\}.$$

2. **Ассоциативность** операции * означает, что $x * (y * z) = (x * y) * z$. Свойство ассоциативности позволяет записывать формулы, содержащие одинаковые ассоциативные связки, без скобок, например, $x \vee y \vee z$, $x \wedge y \wedge z$.
   Логическая операция * ассоциативна, если связка * принадлежит следующему множеству связок (существенно только, чтобы символ в равенстве всюду имел один и тот же смысл):

$$* \in \{\wedge, \vee, +, \equiv\}$$

3. **Дистрибутивность** (распределительный закон) операции * относительно операции $\bullet$ означает, что $x * (y \bullet z) = (x * y) \bullet (x * z)$.

   Дистрибутивность конъюнкции:
   $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ — дистрибутивность конъюнкции относительно дизъюнкции;
   $x \wedge (y + z) = (x \wedge y) + (x \wedge z)$ — дистрибутивность конъюнкции относительно логической суммы.

   Дистрибутивность дизъюнкции:
   $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ — дистрибутивность дизъюнкции относительно конъюнкции;
   $x \vee (y \to z) = (x \vee y) \to (x \vee z)$ —дистрибутивность дизъюнкции относительно импликации;
   $x \vee (y \equiv z) = (x \vee y) \equiv (x \vee z)$ — дистрибутивность дизъюнкции относительно эквивалентности.

   Дистрибутивность импликации:
   $x \to (y \wedge z) = (x \to y) \wedge (x \to z)$ — дистрибутивность импликации относительно конъюнкции;
   $x \to (y \vee z) = (x \to y) \vee (x \to z)$ — дистрибутивность импликации относительно дизъюнкции;
   $x \to (y \to z) = (x \to y) \to (x \to z)$ — дистрибутивность импликации относительно импликации.

4. Имеет место следующее соотношение для двойного отрицания: $\bar{\bar{x}} = x$.

5. Имеют место следующие соотношения между отрицанием, конъюнкцией и дизъюнкцией.
   Правила да Моргана: $\begin{cases} \overline{x \wedge y} = \bar{x} \vee \bar{y} \\ \overline{x \vee y} = \bar{x} \wedge \bar{y} \end{cases}$

   Указанные соотношения отражают отношение двойственности между дизъюнкцией и конъюнкцией.

6. Имеют место следующие соотношения, связанные с «навешиванием отрицания» на элементарные логические функции:
   $\overline{x|y} = x \wedge y$;
   $\overline{x \downarrow y} = x \vee y$;
   $\overline{x + y} = x \equiv y$;
   $\overline{x \equiv y} = x + y$
   $\overline{x \to y} = \bar{x} \vee y = x\bar{y}$.

7. $0 = x \wedge \bar{x} = x \wedge 0 = x + x$
   $1 = x \vee \bar{x} = x \vee 1 = x \equiv x.$

8. Правила поглощения:
   $x \vee (x \wedge y) = x$
   $x \wedge (x \vee y) = x$

9. Выполняются следующие свойства конъюнкции и дизъюнкции:

$$\begin{array}{ll} x \wedge x = x, & x \vee x = x, \\ x \wedge \bar{x} = 0, & x \vee \bar{x} = 1, \\ x \wedge 0 = 0, & x \vee 0 = x, \\ x \wedge 1 = x, & x \vee 1 = 1. \end{array}$$

Все указанные тождества могут быть проверены путем сопоставления функций, реализуемых правой и левой частями формул. □

Введенные нами элементарные функции не являются независимыми, так, например:

$$\begin{aligned} x \equiv y =& (x \to y)\&(y \to x) = x \wedge y \vee \bar{x} \wedge \bar{y}; \\ & x \to y = \bar{x} \vee y. \end{aligned}$$

Имеется гораздо более радикальная возможность сведения: все элементарные функции могут быть выражены через одну-единственную: штрих Шеффера или стрелку Пирса.

Задача. Выразить все элементарные функции через $|$ и $\downarrow$. □

2.4 Разложение функций алгебры логики по переменным

Говоря о языке формул, мы сознательно не касались весьма важного вопроса: всякая ли функция алгебры логики может быть выражена в виде формулы, если допустить некоторый запас элементарных функций? Ближайшие рассмотрения направлены на решение этого вопроса.

Чтобы иметь возможность единообразно записывать переменные с отрицанием и без отрицания введем следующее обозначение:

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

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

Лемма 2.2 (О разложении функции по одной переменной)**.** *Пусть* $f(x_1, \ldots, x_n)$ *— произвольная функция алгебры логики, тогда справедливо следующее представление* f *в форме разложения по переменной* x_1*:*

$$f(x_1, \ldots, x_n) = x_1 \cdot f(1, x_2, \ldots, x_n) \vee \bar{x}_1 \cdot f(0, x_2, \ldots, x_n) \tag{2.1}$$

Доказательство. Отметим прежде всего, что представление (2.1), естественно, справедливо для произвольной переменной x_i из множества переменных функции f. Для доказательства рассмотрим произвольный набор значений переменных $(\alpha_1, \ldots, \alpha_n)$ и покажем, что левая и правая части соотношения (2.1) принимают на нём одно и то же значение.

Рассмотрим набор значений переменных $(1, \alpha_2, \ldots, \alpha_n)$. Левая часть (2.1) принимает на этом наборе значение $f(1, \alpha_2, \ldots, \alpha_n)$, а правая часть — значение $1 \cdot f(1, \alpha_2, \ldots, \alpha_n) \vee 0 \cdot f(0, \alpha_2, \ldots, \alpha_n) = f(1, \alpha_2, \ldots, \alpha_n)$. Таким образом, на наборах $(1, \alpha_2, \ldots, \alpha_n)$ левая и правая части (2.1) принимают одинаковые значения.

Рассмотрим набор значений переменных $(0, \alpha_2, \ldots, \alpha_n)$. Левая часть (2.1) принимает на этом наборе значение $f(0, \alpha_2, \ldots, \alpha_n)$, а правая часть — значение $0 \cdot f(1, \alpha_2, \ldots, \alpha_n) \vee 1 \cdot f(0, \alpha_2, \ldots, \alpha_n) = f(0, \alpha_2, \ldots, \alpha_n)$. Таким образом, на наборах $(0, \alpha_2, \ldots, \alpha_n)$ левая и правая части (2.1) принимают одинаковые значения.

Тем самым мы доказали, что левая и правая части соотношения (2.1) принимают одинаковые значения на всех наборах $(\alpha_1, \ldots, \alpha_n)$. □

Лемма 2.3. *Конъюнкция (произведение)* $x_1^{\sigma_1} K x_n^{\sigma_n} = 1$ *тогда и только тогда, когда* $(x_1, K, x_n) = (\sigma_1, K, \sigma_n)$.

Доказательство. Произведение (конъюнкция) равно 1 тогда и только тогда, когда каждый сомножитель равен 1, но $x^\sigma = 1$ тогда и только тогда, когда $x = \sigma$. □

В дальнейшем будем употреблять следующие обозначения:

$$\bigwedge_{i=1}^{k} = x_1 \wedge x_2 \wedge \ldots \wedge x_k = x_1 x_2 \ldots x_k, \quad \bigvee_{i=1}^{k} = x_1 \vee x_2 \vee \ldots \vee x_k$$

Эти записи имеют смысл и при $k = 1$.

Теорема 2.4 (О разложении функции по нескольким переменным)**.** *Пусть* $f(x_1, \ldots, x_n)$ *— произвольная функция алгебры логики. Тогда ее можно представить в следующей форме:*

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_k)} x_1^{\sigma_1} \ldots x_k^{\sigma_k} \cdot f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n) \tag{2.2}$$

Доказательство. Рассмотрим произвольный набор значений переменных $(\alpha_1, \ldots, \alpha_n)$ и покажем, что левая и правая части соотношения (2.2) принимают на нем одно и то же значение. Левая часть дает $f(\alpha_1, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n)$. Правая часть дает

$$\bigvee_{(\sigma_1, \ldots, \sigma_k)} x_1^{\sigma_1} K x_k^{\sigma_k} \cdot f(\sigma_1, K, \sigma_k, x_{k+1}, K, x_n) =$$
$$= \alpha_1^{\sigma_1} K \alpha_k^{\sigma_k} \cdot f(\alpha_1, K, \alpha_k, \alpha_{k+1}, K, \alpha_n) = f(\alpha_1, K, \alpha_n).$$

□

Представление (2.2) называется дизъюнктивным разложением функции по k переменным.

Пример. Для $k = 2$ разложение в дизъюнктивную форму имеет вид:

$$\begin{aligned} f(x_1, \ldots, x_n) = {} & \bar{x}_1 \bar{x}_2 \cdot f(0, 0, x_3, \ldots, x_n) \vee \\ & \vee \bar{x}_1 x_2 \cdot f(0, 1, x_3, \ldots, x_n) \vee \\ & \vee x_1 \bar{x}_2 \cdot f(1, 0, x_3, \ldots, x_n) \vee \\ & \vee x_1 x_2 \cdot f(1, 1, x_3, \ldots, x_n). \end{aligned}$$

Выпишем такое разложение для конкретной функции трех переменных по переменным x_2 и x_3:

$$\begin{aligned} (x_1 \to x_2) \equiv (x_2 \to x_3) = {} & \bar{x}_2 \bar{x}_3 \cdot ((x_1 \to 0) \equiv (0 \to 0)) \vee \\ & \vee \bar{x}_2 x_3 \cdot ((x_1 \to 0) \equiv (0 \to 1)) \vee \\ & \vee x_2 \bar{x}_3 \cdot ((x_1 \to 1) \equiv (1 \to 0)) \vee \\ & \vee x_2 x_3 \cdot ((x_1 \to 1) \equiv (1 \to 1)). \end{aligned}$$

В качестве следствий получаем два специальных разложения.

1. Разложение по одной переменной, выписанное ранее.
2. Разложение по всем n переменным.

Если $k = n$, то получаем разложение

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

Оно может быть преобразовано при $f(x_1, \ldots, x_n) \neq 0$ следующим образом:

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

Итак, в этом случае разложение имеет вид:

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

Это разложение называется совершенной дизъюнктивной нормальной формой (совершенная ДНФ.). Оно определено для любой функции f, не равной константе 0.

Теорема 2.5. *Произвольную функцию алгебры логики можно выразить формулой при помощи операций* $\wedge$, $\vee$, $\neg$, *причем операция* $\neg$ *применяется только к переменным.*

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

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

2. Пусть $f(x_1, \ldots, x_n) \neq 0$. Представим ее в форме совершенной ДНФ:

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

Таким образом, в обоих случаях функция f выражается в виде формулы через конъюнкцию, дизъюнкцию и отрицание, причем отрицание применяется только к символам переменных. □

Итак, оказалось, что любую булеву функцию можно выразить формулой над множеством операций $\{\vee, \wedge, \neg\}$, состоящим из трех функций: отрицания, конъюнкции и дизъюнкции. Данная теорема носит конструктивный

характер, так как она позволяет для каждой функции построить реализующую ее формулу (совершенную ДНФ). А именно, берем таблицу для функции $f(x_1, \ldots, x_n)$ $(f \neq 0)$ и отмечаем в ней все строки $(\sigma_1, \ldots, \sigma_n)$, в которых $f(\sigma_1, \ldots, \sigma_n) = 1$, для каждой такой строки образуем логическое произведение $x_1^{\sigma_1} \wedge \ldots \wedge x_n^{\sigma_n}$, а затем соединяем все полученные конъюнкции знаком дизъюнкции.

Пример Построить совершенную ДНФ для функции, заданной таблицей:

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	0
0	0	1	1	1	0	1	1
0	1	0	0	1	1	0	0
0	1	1	0	1	1	1	1

Имеем:

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

Если в таблице значений функции много 1 и мало 0, то целесообразно строить функцию по-другому. Совершенная ДНФ есть выражение типа « $\vee$ &», то есть логическая сумма произведений $x_i^{\sigma_i}$. Спрашивается нельзя ли для функции алгебры логики получить разложение типа «&, $\vee$»?

Аналогично только что проведенным доказательствам легко получить, что:

- $x^{\bar{\sigma}} = 0$ тогда и только тогда, когда $x = \sigma$;
- $x_1^{\bar{\sigma}_1} \vee \ldots \vee x_n^{\bar{\sigma}_n}$ обращается в 0 только на наборе $(x_1, \ldots, x_n) = (\sigma_1, \ldots, \sigma_n)$;
- имеет место следующее разложение в конъюнктивную нормальную форму по одной переменной $f(x_1, \ldots, x_n) = (x_1 \vee f(0, x_2, \ldots, x_n)) \cdot \cdot (\bar{x}_1 \vee f(1, x_2, \ldots, x_n))$
- имеет место следующее представление функции в виде совершенной конъюнктивной нормальной формы (совершенная КНФ) для $f \neq 1$:

$$f(x_1, K, x_n) = \bigwedge_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=0}} x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee K \vee x_n^{\bar{\sigma}_n}$$

Использование совершенной КНФ позволяет упростить запись формулы для функции, таблица значений которой содержит мало нулей. □

Кроме отмеченных конъюнктивного и дизъюнктивного разложений функции по переменной часто используется и разложение, основанное на операции логической суммы:

$$f(x_1,\ x_2,\ \ldots,\ x_n) = x_1 \cdot f(1,\ x_2,\ \ldots,\ x_n) + \bar{x}_1 \cdot f(0,\ x_2,\ \ldots,\ x_n)$$

Последовательное применение такого разложения ко всем переменным позволяет выразить произвольную функцию алгебры логики через элементарные функции $\bar{x}$, $x + y$, $x \wedge y$ или, используя соотношение $\bar{x} = x + 1$, лишь через функции $x + y$, $x \wedge y$, 1.

2.5 Функциональная полнота систем функций алгебры логики

Выше мы видели, что всякая функция алгебры логики может быть выражена в виде формулы через элементарные функции $\bar{x}$,$x \wedge y$, $x \vee y$. В связи с этим возникает вопрос, какими свойствами должна обладать система функций, чтобы через функции этой системы можно было выразить произвольную функцию алгебры логики? Мы собираемся дать достаточно исчерпывающий ответ на этот вопрос и показать, что таким свойством обладают и другие системы функций.

Прежде всего уточним, какими средствами из имеющейся системы функций можно получать новые функции. Новые функции получаются из имеющихся в заданной системе функций с помощью операций замены переменных и суперпозиции. Опишем эти две операции.

1. **Замена переменных.**

   Пусть $f(x_1,\ x_2,\ \ldots,\ x_n)$ — заданная функция алгебры логики. Будем говорить, что функция $\varphi(y_1,\ y_2,\ \ldots,\ y_n)$ получена операцией замены переменных из функции $f(x_1,\ x_2,\ \ldots,\ x_n)$, если осуществлена подстановка переменных

   $$s = \begin{pmatrix} x_1 & \ldots & x_n \\ y_1 & \ldots & y_n \end{pmatrix},$$

   то есть вместо каждого вхождения переменой x_1 подставляется переменная y_1, вместо каждого вхождения переменой x_2 подставляется переменная y_2, ..., вместо каждого вхождения переменой x_n подставляется переменная y_n, при этом y_i не обязана отличаться от y_k

при $i \neq k$. Очевидно, что замена переменных включает в себя переименование переменных, перестановку переменных и отождествление переменных.

Пример Пусть имеется функция $f(x_1,\ x_2) = x_1|x_2$. Тогда при замене переменных $\begin{pmatrix} x_1 & x_2 \\ y & y \end{pmatrix}$ из функции $f(x_1,\ x_2)$ можно получить функцию $\varphi(y) = f(y,\ y) = y|y = \bar{y}$.

2. **Суперпозиция функций алгебры логики.**

Определение. Пусть имеется функция $f(x_1,\ x_2,\ \ldots,\ x_n)$ и функции

$$f_1(x_{i_1},\ \ldots,\ x_{m_i}),\ \ i = 1,\ \ldots,\ n,$$

тогда функцию $\varphi = f(f_1(x_{1_1},\ \ldots,\ x_{1_{m_1}}),\ \ldots,\ f_n(x_{n_1},\ \ldots,\ x_{n_{m_n}}))$ будем называть *суперпозицией функции* $f(x_1,\ x_2,\ \ldots,\ x_n)$ *и функций* $f_i(x_{i_1},\ \ldots,\ x_{i_{m_i}}),\ i = 1,\ \ldots,\ n$.

Другими словами: пусть $F = \{f_j\}$ — набор функций алгебры логики, не обязательно конечный. Функция f называется суперпозицией функций из множества F или функцией над F, если она получена из функции $f_j \in F$ путем замены одной или нескольких ее переменных функциями из множества F.

Пример.

Пусть задано множество функций

$$F = \{f_1(x_1),\ f_2(x_1,\ x_2,\ x_3),\ f_3(x_1,\ x_2)\}.$$

Тогда суперпозициями функций из F будут, например, функции:

$$\varphi_1(x_2,\ x_3) = f_3(f_1(x_2),\ f_1(x_3));$$
$$\varphi_2(x_1,\ x_2) = f_2(x_1,\ f_1(x_1),\ f_3(x_1,\ x_2)).$$

Совершенная ДНФ — суперпозиция функций из множества

$$\{x_1 \vee x_2,\ x_1 \wedge x_2,\ \bar{x}\}$$

Определение. Система функций называется *полной*, если при помощи операций суперпозиции и замены переменных из функций этой системы может быть получена любая функция алгебры логики.

Мы уже имеем некоторый набор полных систем:
$\{x \vee y, xy, \bar{x}\}$;
$\{xy, \bar{x}\}$, так как $x \vee y = \overline{\bar{x} \wedge \bar{y}}$;
$\{x \vee y, \bar{x}\}$, так как $xy = \overline{\bar{x} \vee \bar{y}}$
$\{x + y, xy, 1\}$.

Как же определить условия, при которых система полна. С понятием полноты тесно связано понятие замкнутого класса.

2.5.1 Замкнутые классы

Множество (класс) K функций алгебры логики называется *замкнутым классом*, если оно содержит все функции, получающиеся из K операциями суперпозиции и замены переменных, и не содержит никаких других функций.

Пусть K — некоторое подмножество функций из P_2. Замыканием K называется множество всех булевых функций, представимых с помощью операций суперпозиции и замены переменных функций из множества K. Замыкание множества K обозначается через $[K]$.

В терминах замыкания можно дать другие определения замкнутости и полноты (эквивалентные исходным):
K — замкнутый класс, если $K = [K]$;
K — полная система, если $[K] = P_2$.

Примеры.

- $\{0\}$, $\{1\}$ — замкнутые классы.
- Множество функции одной переменной — замкнутый класс.
- $\{x, \bar{x}\}$ — замкнутый класс.
- Класс $\{1, x + y\}$ не является замкнутым классом.

Рассмотрим некоторые важнейшие замкнутые классы.

1. $\mathbf{T_0}$ — класс функций, сохраняющих 0.
   Обозначим через T_0 класс всех функций алгебры логики $f(x_1, x_2, \ldots, x_n)$, сохраняющих константу 0, то есть функций, для которых
   $f(0, \ldots, 0) = 0$.

$$T_0 = \{f(x_1, x_2, \ldots, x_n) | f(0, \ldots, 0) = 0\}.$$

Легко видеть, что есть функции, принадлежащие T_0, и функции, этому классу не принадлежащие:

$$0,\ x,\ xy,\ x \vee y,\ x + y \in T_0$$
$$1,\ \bar{x} \notin T_0$$

Из того, что $\bar{x} \notin T_0$ следует, например, что $\bar{x}$ нельзя выразить через дизъюнкцию и конъюнкцию.

Поскольку таблица для функции f из класса T_0 в первой строке содержит значение 0, то для функций из T_0 можно задавать произвольные значения только на $2n-1$ наборе значений переменных, то есть

$$|T_0^{(n)}| = 2^{2^n-1} = \frac{1}{2}|P_2|,$$

где $T_0^{(n)}$ — множество функций, сохраняющих 0 и зависящих от n переменных.

Покажем, что T_0 — замкнутый класс. Так как $x \in T_0$, то для обоснования замкнутости достаточно показать замкнутость относительно операции суперпозиции, поскольку операция замены переменных есть частный случай суперпозиции с функцией x.

Пусть $f(\tilde{x}_m),\ f_1(\tilde{x}_{k_1}),\ \ldots,\ f_m(\tilde{x}_{k_m}) \in T_0$. Тогда достаточно показать, что $\varphi = f(f_1,\ \ldots,\ f_m) \in T_0$. Последнее вытекает из цепочки равенств

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

2. $\mathbf{T_1}$ — класс функций, сохраняющих 1.

Обозначим через T_1 класс всех функций алгебры логики $f(x_1,\ x_2,\ \ldots,\ x_n)$, сохраняющих константу 1, то есть функций, для которых $f(1,\ \ldots,\ 1) = 1$.

$$T_1 = \{f(x_1,\ x_2,\ \ldots,\ x_n) | f(1,\ \ldots,\ 1) = 1\}$$

Легко видеть, что есть функции, принадлежащие T_1, и функции, этому классу не принадлежащие:

$$1,\ x,\ xy,\ x \vee y,\ x \equiv y \in T_1$$
$$0,\ \bar{x},\ x + y \notin T_1$$

Из того, что $x + y \notin T_0$ следует, например, что $x + y$ нельзя выразить через дизъюнкцию и конъюнкцию.

Результаты о классе T_0 тривиально переносятся на класс T_1. Таким образом, имеем:
T_1 — замкнутый класс;

$$|T_1^{(n)}| = 2^{2^n-1} = \frac{1}{2}|P_2|.$$

3. **L** — класс линейных функций.

Обозначим через L класс всех функций алгебры логики $f(x_1, x_2, \ldots, x_n)$, являющихся линейными:

$$L = \left\{ \begin{array}{ll} f(x_1, x_2, \ldots, x_n) & : f(x_1, x_2, \ldots, x_n) = \alpha_0 + \alpha_1 x_1 + \ldots \\ & \ldots + \alpha_n x_n;\ \alpha_i \in \{0,1\},\ i = 1, \ldots, n \end{array} \right\}$$

Легко видеть, что есть функции, принадлежащие L , и функции, этому классу не принадлежащие:

$$0,\ 1,\ x,\ x+y,\ x_1 \equiv x_2 = x_1 + x_2 + 1,\ \bar{x} = x + 1 \in L;$$
$$xy,\ x \vee y \notin L.$$

Докажем, например, что $x \vee y \notin L$.

Предположим противное. Будем искать выражение для $x \vee y$ в виде линейной функции с неопределенными коэффициентами:

$$x \vee y = \alpha + \beta x + \gamma y$$

При $x = y = 0$ имеем $\alpha = 0$,
при $x = 1,\ y = 0$ имеем $\beta = 1$,
при $x = 0,\ y = 1$ имеем $\gamma = 1$,
но тогда при $x = 1,\ y = 1$ имеем $1 \vee 1 \neq 1 + 1$, что доказывает нелинейность функции $x \vee y$.

Доказательство замкнутости класса линейных функций совершенно очевидно.

Поскольку линейная функция однозначно определяется заданием значений $n + 1$ коэффициента $\alpha_0, \ldots, \alpha_n$, число линейных функций в классе $L^{(n)}$ функций, зависящих от n переменных равно 2^{n+1}.

$$|L^{(n)}| = 2^{n+1}.$$

4. S — класс самодвойственных функций.

Определение класса самодвойственных функций основано на использовании так называемого принципа двойственности и двойственных функций.

Определение. Функция $f^*(\tilde{x}_n)$, определяемая равенством $f^*(\tilde{x}_n) = \bar{f}(\bar{x}_1, \ldots, \bar{x}_n)$ называется *двойственной к функции* $f(\tilde{x}_n)$.

Очевидно, что таблица для двойственной функции (при стандартной упорядоченности наборов значений переменных) получается из таблицы для исходной функции инвертированием (то есть заменой 0 на 1 и 1 на 0) столбца значений функции и его переворачиванием.

Легко видеть, что:

$$0^* = 1,$$

$$1^* = 0,$$

$$x^* = x,$$

$$\bar{x}^* = \bar{x},$$

$$(x_1 \vee x_2)^* = x_1 \wedge x_2,$$

$$(x_1 \wedge x_2)^* = x_1 \vee x_2.$$

Из определения вытекает, что $(f^*)^* = f$, то есть функция f является двойственной к f^*.

Пусть функция выражена с помощью суперпозиции через другие функции. Спрашивается, как построить формулу, реализующую $f^*(\tilde{x}_n)$? Обозначим через $\tilde{x}_n = (x_1, \ldots, x_n)$ все различные символы переменных, встречающиеся в наборах $(x_{11}, \ldots, x_{1p_1}), \ldots, (x_{m1}, \ldots, x_{mp_m})$.

Теорема 2.6. *Если функция φ получена как суперпозиция функций $f, f_1, f_2, \ldots, f_m$, то есть*

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

то

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

функция, двойственная к суперпозиции, есть суперпозиция двойственных функций.

Доказательство.

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

Теорема доказана. □

Из теоремы вытекает принцип двойственности: если формула A реализует функцию $f(x_1,\ \ldots,\ x_n)$, то формула, полученная из A заменой входящих в нее функций на двойственные им, реализует двойственную функцию $f^*(x_1,\ \ldots,\ x_n)$.

Обозначим через S класс всех самодвойственных функций из P_2:

$$S = \{f | f^* = f\}$$

Легко видеть, что есть функции, принадлежащие S, и функции, этому классу не принадлежащие:

$$\begin{aligned}&x,\ \bar{x} \in L;\\ &0,\ 1,\ xy,\ x \vee y \notin L.\end{aligned}$$

Менее тривиальным примером самодвойственной функции является функция

$$h(x, y, z) = xy \vee xz \vee yz;$$

используя теорему о функции, двойственной к суперпозиции, имеем

$$h^*(x,\ y,\ z) = (x \vee y) \wedge (x \vee z) \wedge (y \vee z) = xy \vee xz \vee yz = h^*;\ h \in S.$$

Для самодвойственной функции имеет место тождество:

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

так что 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 стандартной таблицы. Поэтому число самодвойственных функций в классе $S^{(n)}$ функций, зависящих от n переменных, равно:

$$|S^{(n)}| = 2^{2^{n-1}}.$$

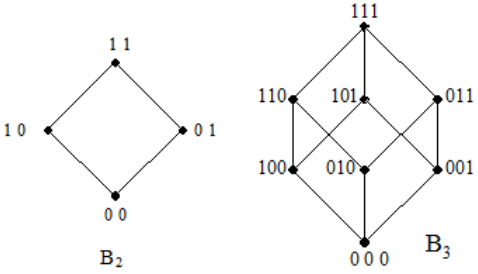
Докажем теперь, что класс S замкнут. Так как $x \in S$, то для обоснования замкнутости достаточно показать замкнутость относительно операции суперпозиции, поскольку операция замены переменных есть частный случай суперпозиции с функцией x. Пусть $f(\tilde{x}_m)$, $f_1(x_{k_1}), \ldots, f_m(\tilde{x}_{k_m}) \in S$. Тогда достаточно показать, что $\varphi = f(f_1, \ldots, f_m) \in S$. Последнее устанавливается непосредственно:

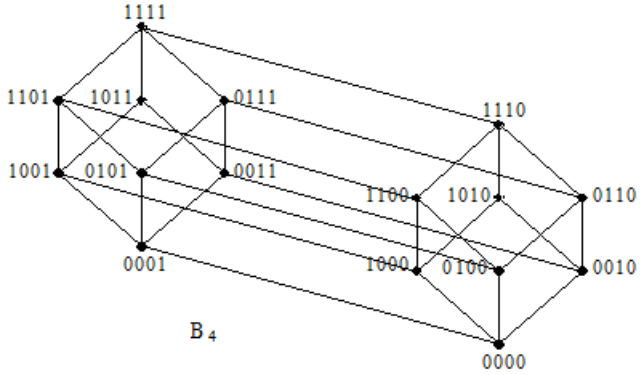
$$\varphi^* = f^*(f_1^*, \ldots, f_m^*) = f(f_1, \ldots, f_m).$$

5. **M** — класс монотонных функций.

Прежде чем определять понятие монотонной функции алгебры логики, необходимо ввести отношение упорядоченности на множестве наборов ее переменных.

Говорят, что набор $\tilde{\alpha} = \tilde{\alpha}_n = (\alpha_1, \ldots, \alpha_n)$ предшествует набору $\tilde{\beta} = \tilde{\beta}_n = (\beta_1, \ldots, \beta_n)$ (или «не больше $\tilde{\beta}$», или «меньше или равен $\tilde{\beta}$»), и применяют обозначение $\tilde{\alpha} \leq \tilde{\beta}$, если $\alpha_i \leq \beta_i$ для всех $i = 1, \ldots, n$. Если $\tilde{\alpha} \leq \tilde{\beta}$ и $\tilde{\alpha} \neq \tilde{\beta}$, то будем говорить, что набор $\tilde{\alpha}$ строго предшествует набору $\tilde{\beta}$ (или «строго меньше», или «меньше» набора $\tilde{\beta}$), и использовать обозначение $\tilde{\alpha} < \tilde{\beta}$. Наборы $\tilde{\alpha}$ и $\tilde{\beta}$ называются сравнимыми, если либо $\tilde{\alpha} \leq \tilde{\beta}$, либо $\tilde{\beta} \leq \tilde{\alpha}$. В случае, когда ни одно из этих соотношений не выполняется, наборы $\tilde{\alpha}$ и $\tilde{\beta}$ называются несравнимыми. Например, $(0,1,0,1) \leq (1,1,0,1)$, но наборы $(0,1,1,0)$ и $(1,0,1,0)$ несравнимы. Тем самым отношение $\leq$ (его часто называют отношением предшествования) является частичным порядком на множестве B_n. Ниже приведены диаграммы частично упорядоченных множеств B_2, B_3 и B_4.





Введенное отношение частичного порядка — исключительно важное понятие, далеко выходящее за рамки нашего курса.

Теперь мы имеем возможность определить понятие монотонной функции.

Функция алгебры логики $f(\tilde{x}_n)$ называется *монотонной*, если для любых двух наборов $\tilde{\alpha}_n$ и $\tilde{\beta}_n$, таких, что $\tilde{\alpha}_n \le \tilde{\beta}_n$, имеет место неравенство $f(\tilde{\alpha}_n) \le f(\tilde{\beta}_n)$. Множество всех монотонных функций алгебры логики обозначается через M, а множество всех монотонных функций, зависящих от n переменных — через $M^{(n)}$.

Легко видеть, что есть функции, принадлежащие M, и функции, этому классу не принадлежащие:

$$0,\ 1,\ x,\ xy,\ x \vee y \in M;$$
$$x + y,\ x \to y,\ x \equiv y \notin M.$$

Покажем, что класс монотонных функций M — замкнутый класс. Так как $x \in M$, то для обоснования замкнутости достаточно показать замкнутость относительно операции суперпозиции, поскольку операция замены переменных есть частный случай суперпозиции с функцией x.

Пусть $f(\tilde{x}_m), f_1(\tilde{x}_{k_1}),\ \ldots,\ f_m(\tilde{x}_{k_m}) \in M$. Тогда достаточно показать, что $\varphi = f(f_1,\ \ldots,\ f_m) \in M$.

Пусть $\tilde{x} = \tilde{x}_n = (x_1,\ \ldots,\ x_n),\ \tilde{x}_{k_1} = (x_{11},\ \ldots,\ x_{1k_1}),\ \ldots,$
$\tilde{x}_{k_m} = (x_{m1},\ \ldots,\ x_{mk_m})$ — наборы переменных, соответственно, функций $\varphi, f_1,\ \ldots,\ f_m$, причем множество переменных функции φ состоит из тех и только тех переменных, которые встречаются у функций

$f_1, \ldots, f_m$. Пусть $\tilde{\alpha}$ и $\tilde{\beta}$ — два набора значений переменной $\tilde{x}$, причем $\tilde{\alpha} \leq \tilde{\beta}$. Эти наборы определяют наборы $\tilde{\alpha}_{k_1}, \tilde{\beta}_{k_1}, \ldots, \tilde{\alpha}_{k_m}, \tilde{\beta}_{k_m}$ значений переменных $\tilde{x}_{k_1}, \ldots, \tilde{x}_{k_m}$, такие, что $\tilde{\alpha}_{k_1} \leq \tilde{\beta}_{k_1}, \ldots, \tilde{\alpha}_{k_m} \leq \tilde{\beta}_{k_m}$. В силу монотонности функций $f_1, \ldots, f_m$

$$f_1(\tilde{\alpha}_{k_1}) \leq f_1(\tilde{\beta}_{k_1}), \ldots, f_m(\tilde{\alpha}_{k_m}) \leq f_m(\tilde{\beta}_{k_m}),$$

поэтому

$$(f_1(\tilde{\alpha}_{k_1}), \ldots, f_m(\tilde{\alpha}_{k_m})) \leq (f_1(\tilde{\beta}_{k_1}), \ldots, f_m(\tilde{\beta}_{k_m})),$$

и в силу монотонности функции f

$$f(f_1(\tilde{\alpha}_{k_1}), \ldots, f_m(\tilde{\alpha}_{k_m})) \leq f(f_1(\tilde{\beta}_{k_1}), \ldots, f_m(\tilde{\beta}_{k_m})).$$

Отсюда получаем

$$\varphi(\tilde{\alpha}) = f(f_1(\tilde{\alpha}_{k_1}), \ldots, f_m(\tilde{\alpha}_{k_m})) \leq f(f_1(\tilde{\beta}_{k_1}), \ldots, f_m(\tilde{\beta}_{k_m})) = \varphi(\tilde{\beta}) \ \square$$

Число монотонных функций, зависящих от n переменных, точно неизвестно. Легко может быть получена оценка снизу:

$$|M^{(n)}| \geq 2^{\binom{n}{[n/2]}},$$

где $[n/2]$— есть целая часть от $n/2$.

Так же просто получается слишком завышенная оценка сверху:

$$|M^{(n)}| \leq 2 + n^{\binom{n}{[n/2]}}.$$

Уточнение этих оценок — важная и интересная задача современных исследований.

2.5.2 Критерий полноты

Теперь мы в состоянии сформулировать и доказать критерий полноты (теорему Поста), определяющий необходимые и достаточные условия полноты системы функций. Предварим формулировку и доказательство критерия полноты несколькими необходимыми леммами, имеющими и самостоятельный интерес.

Лемма 2.7 (Лемма о несамодвойственной функции.)**.** *Если $f(x_1, \ldots, x_n) \notin S$, то из нее путем подстановки функций x и $\bar{x}$ можно получить константу.*

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

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

Заменим аргументы в функции f:

$$x_i \text{ заменяется на } \begin{cases} x, & \text{если } \alpha_i = 1 \text{ в наборе } \alpha; \\ \bar{x}, & \text{если } \alpha_i = 0 \text{ в наборе } \alpha, \end{cases}$$

то есть положим $x_i = x^{\alpha_i}$, и рассмотрим функцию

$$\varphi(x) = f(x^{\alpha_1}, x^{\alpha_2}, \ldots, x^{\alpha_n}).$$

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

Тем самым мы получили константу (правда, неизвестно, какая это константа: 0 или 1). □

Лемма 2.8 (Лемма о немонотонной функции.)**.** *Если функция $f(x_1, \ldots, x_n)$ немонотонна, $f(x_1, \ldots, x_n) \notin M$, то из нее путем замены переменных и подстановки констант 0 и 1 можно получить отрицание.*

Доказательство. Так как $f(x_1, \ldots, x_n) \notin M$, то найдутся наборы $\tilde{\alpha}$ и $\tilde{\beta}$ значений ее переменных, $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$, $\tilde{\beta} = (\beta_1, \ldots, \beta_n)$, такие что $\tilde{\alpha} \leq \tilde{\beta}$ и $f(\tilde{\alpha}) > f(\tilde{\beta})$, причем хотя бы для одного значения i имеет место $\alpha_i < \beta_i$. Выполним следующую замену переменных функции f:

$$x_i \text{ заменим на } \begin{cases} 0, & \text{если } \alpha_i = \beta_i = 0; \\ 1, & \text{если } \alpha_i = \beta_i = 1; \\ x, & \text{если } \alpha_i < \beta_i. \end{cases}$$

После такой подстановки получим функцию одной переменной $\varphi(x)$, для которой имеем:

$$\varphi(0) = f(\bar{\alpha}) = 1;$$
$$\varphi(1) = f(\bar{\beta}) = 0.$$

Это означает, что $\varphi(x) = \bar{x}$. Лемма доказана. □

Лемма 2.9 (Лемма о нелинейной функции.). *Если* $f(x_1, \ldots, x_n) \notin L$, *то из нее путем подстановки констант 0, 1 и использования функции* $\bar{x}$ *можно получить функцию* $x_1 \& x_2$.

Доказательство. Представим f в виде ДНФ (например, совершенной ДНФ) и воспользуемся соотношениями:

$$x_1 \vee x_2 = \overline{x_1 \wedge x_2};$$
$$\bar{x} = x + 1;$$
$$K \wedge K = K;$$
$$K + K = 0.$$

Пример. Приведем два примера применения указанных преобразований.

$$x_1 \vee x_2 = (x_1 + 1)(x_2 + 1) + 1 = x_1x_2 + x_1 + x_2;$$
$$x_1x_2 \vee x_3x_4 = (x_1x_2 + 1)(x_3x_4 + 1) + 1 = x_1x_2x_3x_4 + x_2x_3 + x_3x_4.$$

Таким образом, функция, записанная в дизъюнктивной нормальной форме, после применения указанных соотношений, раскрытия скобок и несложных алгебраических преобразований переходит в полином по mod 2 (полином Жегалкина):

$$f = \sum_{i_1, \ldots, i_s} \alpha_{i_1, \ldots, i_s} x_{i_1} x_{i_2} \ldots x_{i_s} =$$
$$= \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n +$$
$$+ \alpha_{1,2} x_1 x_2 + \alpha_{1,3} x_1 x_3 + \ldots + \alpha_{n-1,n} x_{n-1} x_n +$$
$$+ \alpha_{1,2,3} x_1 x_2 x_3 + \ldots + \alpha_{n-2,n-1,n} x_{n-2} x_{n-1} x_n + \ldots$$
$$+ \alpha_{1,2,\ldots,n} x_1 x_2 \ldots x_n = A_0 + A_1 + \ldots + A_r,$$

где A_0 константа, а A_i — конъюнкция некоторых переменных из числа $x_1, \ldots, x_n$, $i = 1, 2, \ldots, r$.

Если каждая конъюнкция A_i состоит лишь из одной переменной, то f — линейная функция, что противоречит условию леммы.

Следовательно, в полиноме Жегалкина для функции f найдется член, в котором содержится не менее двух сомножителей. Без ограничения общности можно считать, что среди этих сомножителей присутствуют переменные x_1 и x_2. Тогда полином можно преобразовать следующим образом:

$$f = x_1x_2f_1(x_3, \ldots, x_n) + x_1f_2(x_3, \ldots, x_n) + x_2f_3(x_3, \ldots, x_n) + f_4(x_3, \ldots, x_n),$$

где $f_1(x_3,\ \ldots,\ x_n) \neq 0$ (в противном случае в полином не входит конъюнкция, содержащая конъюнкцию x_1x_2).

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.

Воспользуемся операцией отрицания, которая у нас имеется, и рассмотрим функцию $\psi(x_1,\ x_2)$, получающуюся 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

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

Следовательно,

$$\psi(x_1, x_2) = x_1x_2.$$

Лемма доказана полностью. □

Лемма 2.10 (Основная лемма критерия полноты.)**.** *Если в классе $F = \{f\}$ функций алгебры логики содержатся функции, не сохраняющие единицу, не сохраняющие 0, несамодвойственные и немонотонные:*

$$f_{\bar{0}} \notin T_0,\ f_{\bar{1}} \notin T_1,\ f_{\bar{S}} \notin S,\ f_{\bar{M}} \notin M,$$

то из функций этой системы операциями суперпозиции и замены переменных можно получить константы 0, 1 и функцию $\bar{x}$.

Доказательство. Рассмотрим функцию $f_{\bar{0}} \notin T_0$. Тогда

$$f_{\bar{0}}(0,\ 0,\ \ldots,\ 0) = 1.$$

Возможны два случая последующих рассмотрений, в дальнейшем изложении обозначенные как 1) и 2).

1). Функция $f_{\bar{0}}$ на единичном наборе принимает значение 0:

$$f_{\bar{0}}(1,\ 1,\ \ldots,\ 1) = 0.$$

Заменим все переменные функции $f_{\bar{0}}$ переменной x.

Тогда функция

$$\varphi(x) = f_{\bar{0}}(x, \ldots, x)$$

есть $\bar{x}$, ибо

$$\varphi(0) = f_{\bar{0}}(0, \ldots, 0) = 1 \text{ и } \varphi(1) = f_{\bar{0}}(1, \ldots, 1) = 0.$$

Возьмем несамодвойственную функцию $f_{\bar{S}}$. Так как функцию $\bar{x}$ мы уже получили, то по лемме о несамодвойственной функции (лемма 2.7) из $f_{\bar{S}}$ можно получить константу. Вторую константу можно получить из первой, используя функцию $\bar{x}$. Итак, в первом рассмотренном случае получены константы и отрицание.

2). Функция $f_{\bar{0}}$ на единичном наборе принимает значение 1:

$$f_{\bar{0}}(1, 1, \ldots, 1) = 1.$$

Заменим все переменные функции $f_{\bar{0}}$ переменной x (отождествим все переменные). Тогда функция $\varphi(x) = f_{\bar{0}}(x, \ldots, x)$ есть константа 1, ибо

$$\varphi(0) = f_{\bar{0}}(0, \ldots, 0) = 1 \text{ и } \varphi(1) = f_{\bar{0}}(1, \ldots, 1) = 1.$$

Вторая константа 0 получается из функции $f_{\bar{1}}$, не сохраняющей единицу, $f_{\bar{1}} \notin T_1 : f_{\bar{1}}(1, 1, \ldots, 1) = f_{\bar{1}}(\varphi(x), \varphi(x), \ldots, \varphi(x)) = 0$ Теперь на основании леммы о немонотонной функции (лемма 2.8) из имеющейся у нас немонотонной функции $f_{\bar{M}}$ и полученных констант 0 и 1 можно получить отрицание $\bar{x}$. Второй случай, а вместе с ним и основная лемма критерия полноты, полностью доказаны. □

Теорема 2.11 (Критерий полноты систем функций алгебры логики (теорема Поста))**.** *Для того, чтобы система функций $F = \{f_i\}$ была полной, необходимо и достаточно, чтобы она целиком не содержалась ни в одном из пяти замкнутых классов T_0, T_1, L, S, M, то есть для каждого из классов T_0, T_1, L, S, M в F найдется хотя бы одна функция, этому классу не принадлежащая.*

Доказательство. **Необходимость**. Пусть F — полная система. Допустим, что F содержится в одном из указанных классов, обозначим его через K, т.е. $F \subseteq K$. Последнее включение невозможно, так как K — замкнутый класс, не являющийся полной системой.

Достаточность. Пусть система функций $F = \{f_i\}$ целиком не содержится ни в одном из пяти замкнутых классов T_0, T_1, L, S, M. Возьмем в F функции:

$$f_{\bar{0}} \notin T_0,\ f_{\bar{1}} \notin T_1,\ f_{\overline{L}} \notin L,\ f_{\overline{S}} \notin S,\ f_{\overline{M}} \notin M.$$

Тогда на основании основной леммы (лемма 2.10.) из функции не сохраняющей 0, функции не сохраняющей 1, несамодвойственной и немонотонной функций можно получить константы 0, 1 и функцию отрицание $\bar{x}$:

$$\{f_{\bar{0}},\ f_{\bar{1}},\ f_{\bar{S}},\ f_{\bar{M}}\} \Rightarrow \{0,\ 1,\ \bar{x}\}.$$

На основании леммы о нелинейной функции (лемма) из констант, отрицания и нелинейной функции можно получить конъюнкцию:

$$\{0,\ 1,\ \bar{x},\ f_{\bar{L}}\} \Rightarrow \{x_1 \wedge x_2\}.$$

Система функций $\{\bar{x},\ x_1 \wedge x_2\}$ — полная система по теореме о возможности представления любой функции алгебры логики в виде совершенной дизъюнктивной нормальной формы (заметим, что дизъюнкция может быть выражена через конъюнкцию и отрицание в виде $x_1 \vee x_2 = \overline{\bar{x}_1 \wedge \bar{x}_2}$).

Теорема доказана полностью. □

Примеры.

1. Покажем, что функция $f(x, y) = x|y$ образует полную систему. Построим таблицу значений функции $x|y$:

x	y	$x\|y$
0	0	1
0	1	1
1	0	1
1	1	0

$f(0,\ 0) = 1$, следовательно, $x|y \notin T_0$.

$f(1,\ 1) = 0$, следовательно, $x|y \notin T_1$.

$f(0,\ 0) = 1, f(1, 1) = 0$, следовательно, $x|y \notin M$

$f(0,\ 1) = f(1,\ 0) = 1$, — на противоположных наборах $x|y$ принимает одинаковые значения, следовательно $x|y \notin S$.

Наконец, $x|y = \overline{x \wedge y} = xy + 1$, что означает нелинейность функции $x|y$.

На основании критерия полноты можно утверждать, что $f(x, y) = x|y$ образует полную систему. □

2. Покажем, что система функций $\{\bar{x}, x \to y\}$ образует полную систему.

Действительно,

$$\bar{x} \notin T_0,\ \bar{x} \notin T_1,\ \bar{x} \notin M,\ (x \to y) \notin S,\ x \to y = \bar{x} \vee y = (xy + x + 1) \notin L.$$

Тем самым среди функций нашей системы найдены: функция, не сохраняющая 0, функция, не сохраняющая 1, несамодвойственная, немонотонная и

нелинейная функции. На основании критерия полноты можно утверждать, что система функций $\{\bar{x}, x \to y\}$ образует полную систему. □

Таким образом мы убедились, что критерий полноты дает конструктивный и эффективный способ выяснения полноты систем функций алгебры логики.

Сформулируем теперь три следствия из критерия полноты.

Следствие 1. Всякий замкнутый класс K функций алгебры логики, не совпадающий со всем множеством функций алгебры логики ($K \neq P_2$), содержится по крайней мере в одном из построенных замкнутых классов.

Определение. Замкнутый класс K называется *предполным*, если K неполный и для любой функции $f \notin K$ класс $K \cup \{f\}$ — 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, L, M, S.

Для доказательства следствия нужно проверить только то, что ни один из этих классов не содержится в другом, что подтверждается, например, следующей таблицей принадлежности функций различным классам:

	T_0	T_1	L	S	M
0	+	−	+	−	+
1	−	+	+	−	+
$\bar{x}$	−	−	+	+	−

Следствие 3. Из всякой полной системы функций можно выделить пол-ную подсистему, содержащую не более четырех функций.

Из доказательства критерия полноты следует, что можно выделить не более пяти функций. Из доказательства основной леммы (лемма 2.10) следует, что $f_{\bar{0}}(x,\ x,\ \ldots,\ x) \notin T_0$ либо несамодвойственна, либо не сохраняет единицу и не монотонна. Поэтому нужно не более четырех функций.

2.5.3 Представление о результатах Поста

Весьма глубокое изучение замкнутых классов в P_2 было осуществлено американским математиком Э. Постом в 1921 — 1941 годах. Им была описана структура всех замкнутых классов в P_2. Сформулируем некоторые из важнейших результатов этих исследований.

Определение. Система функций $\{f_1,\ f_2,\ \ldots,\ f_n,\ \ldots\}$ из замкнутого класса K называется полной в K, если ее замыкание совпадает с K. □

Иначе говоря, система полна в K, если каждая функция из K может быть выражена в виде формулы через функции данной системы.

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

Так, система $f_1 = x_1x_2, f_2 = 0, f_3 = 1, f_4 = x_1 + x_2$ является базисом в P_2. Можно показать, что система функций $\{0,\ 1,\ x_1x_2,\ x_1 \vee x_2\}$ является базисом для класса M монотонных функций.

Теорема 2.12. *Каждый замкнутый класс из P_2 имеет конечный базис.*

Теорема 2.13. *Мощность множества замкнутых классов в P_2 счетная.*

Хотя вторая из приведенных теорем логически вытекает из первой, однако в доказательствах Поста сначала устанавливается второй факт, а затем — первый.

Глава 3

Элементы теории графов

Начало теории графов как математической дисциплине было положено Эйлером в его знаменитом рассуждении о Кёнигсбергских мостах. Однако эта статья Эйлера 1736 года была единственной на эту тему в течение почти ста лет. Естественные науки оказали свое влияние на развитие теории графов благодаря исследованиям электрических сетей, моделей кристаллов и структур молекул. Развитие формальной логики привело к изучению бинарных отношений в форме графов. Большое число популярных головоломок поддавалось формулировке непосредственно в терминах графов, и это приводило к пониманию, что многие задачи такого рода содержат некоторое математическое ядро, важность которого выходит за рамки конкретного вопроса. Наиболее знаменитая из этих задач — проблема четырех красок, впервые поставленная перед математиками Де Морганом около 1850 г. Никакая другая проблема не вызывала столь многочисленных и остроумных работ в области теории графов. Благодаря своей простой формулировке и раздражающей неуловимости она до 1973 г. оставалась мощным стимулом исследований различных свойств графов (предложенное в 1973 году положительное решение этой проблемы до сих пор оспаривается некоторыми математиками).

Настоящее столетие было свидетелем неуклонного развития теории графов, которая за последние двадцать лет вступила в новый период интенсивных разработок. В этом процессе явно заметно влияние запросов новых областей приложений: теории игр, программирования, оптимизации, теории передачи сообщений, электрических и контактных сетей, а также проблем биологии и психологии.

Предметом первых задач в теории графов были конфигурации, состоящие из точек и соединяющих их линий. В этих рассмотрениях было несущественно, прямые ли это линии или же они являются криволинейными непрерывными дугами, соединяющими концевые точки, где расположены эти линии, являются ли они длинными или короткими. Существенно лишь то, что они соединяют две данные точки. Это приводит в определению графа как абстрактного математического понятия.

Определение. *Граф* $G =< V,\ E >$ есть совокупность множества *вершин* V и множества *ребер (дуг)* E, причем $E \subseteq V \times V$ есть множество упорядоченных пар $< x,\ y >$ для ориентированного графа и $E \subseteq \{\{x,\ y\} : (x,\ y \in V) \wedge (x \neq y)\}$ для неориентированного графа (при этом для неориентированного графа считаем, что ребра $\{a,\ b\}$ и $\{b,\ a\}$ совпадают $\{a,\ b\} = \{b,\ a\}$).

Определение. Граф *неориентированный*, если все его ребра не ориентированы, и граф *ориентированный*, если все его ребра ориентированы.

Ребро ориентированного графа мы обозначаем символом $< x,\ y >$; ребро неориентированного графа обозначается символом $\{x,\ y\}$. В случае, когда несущественно, о каком ребре идет речь, или когда это ясно из контекста, будем произвольное ребро графа обозначать символом $(x,\ y)$.

В приложениях граф обычно интерпретируется как совокупность точек V, в которой точки x и y соединены дугой со стрелкой если $< x,\ y > \in E$ и дугой без стрелки если $\{x,\ y\} \in E$. В случае ориентированного графа для ребра $< x,\ y >$ вершина x — *начальная* вершина, y — *конечная* вершина ребра. Можно также говорить, что $e =< x,\ y >$ есть ребро, *выходящее* из вершины x (*исходящее* из вершины x) и *входящее* в вершину y. Как в случае ориентированного, так и в случае неориентированного ребра говорят, что ребро e *инцидентно* вершинам x и y, а так же, что x и y инцидентны ребру e. Говорят, что две вершины x и y графа *смежны*, если $(x,\ y)$ является ребром. Два ребра *смежны*, если они имеют общую вершину.

При фактическом изображении графа имеется большая свобода в размещении вершин и в выборе формы соединяющих их дуг. Поэтому может оказаться, что один и тот же граф представляется совершенно различными чертежами.

Определение. Будем говорить, что два графа $G =< V,\ E >$ и $G^\circ =< V^\circ,\ E^\circ >$ *изоморфны*, если существует такое взаимно-однозначное соответствие между множествами их вершин V и V°, что вершины соединены ребрами в одном графе тогда и только тогда, когда соответствующие им вершины соединены в другом графе.

Таким образом, если $V \Leftrightarrow V'$ и при этом соответствии $x \Leftrightarrow x'$, $y \Leftrightarrow y'$, то $(x,\ y) \in E$ тогда и только тогда, когда $(x',\ y') \in E'$; $((x,\ y) \in E \Leftrightarrow (x',\ y') \in E')$.

Задача.

Доказать, что графы, изображенные на следующем рисунке 3.1, изоморфны.

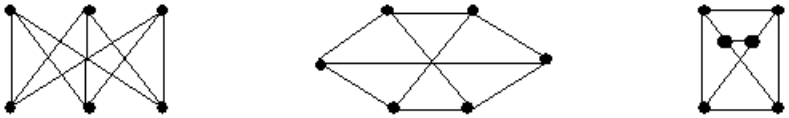


Рис. 3.1

Вершина, не инцидентная никакому ребру, называется изолированной. При определении множества вершин V данного графа часто имеет смысл учитывать только неизолированные вершины.

Важным случаем является неориентированный полный граф $U = U(V)$, ребрами которого являются всевозможные пары $\{x,\ y\}$ для всех различных вершин x и y из V. В ориентированном полном графе $U^{(d)}(V)$ имеются пары ребер $< x,\ y >$ и $< y,\ x >$ для всех различных вершин $x,\ y \in V$.

Сформулированное определение графа, вместе с соответствующим изображением, достаточно для многих задач. Однако, для некоторых целей желательно понятие графа несколько расширить.

1. Можно допускать рёбра, у которых обе вершины совпадают $l = (x,\ x)$. Такое ребро называется петлей. U_0 — полный граф с петлями.

2. Пара вершин может соединяться несколькими ребрами $e_i = (x,\ y)_i$, в частности вершины x и y могут соединяться несколькими ребрами в каждом направлении.

3.1 Степени вершин

Граф называется *конечным*, если число его ребер конечно. При таком определении конечный граф может иметь бесконечное число вершин, но все они, кроме конечного числа, изолированные.

Пусть G — неориентированный граф. Число $\rho(x)$ ребер, инцидентных вершине x, называется локальной степенью, или просто степенью вершины x графа G. В каждом случае должно быть указано, считается петля однократной или двойной.

Пусть $\rho(a,\ b) = \rho(b,\ a)$ — число ребер, соединяющих вершины a и b.

Очевидно, каждая степень каждой вершины есть сумма кратностей в вершине a:

$$\rho(a) = \sum_{b \in V} \rho(a,\ b)$$

Обозначим через $n_e = n_e(G)$ число ребер в неориентированном графе G.

Так как каждое ребро учитывается в двух степенях в вершинах a и b, то $2n_e = \sum\limits_{a \in V} \rho(a)$. Формула остается справедливой и при наличии петель, если только в локальных степенях вершин считать их дважды. Поэтому

$$2n_e = \sum_{a,\ b \in V} \rho(a,\ b)$$

Отсюда следует

Теорема 3.1. *В конечном графе число вершин нечетной степени четно.*

3.2 О машинном представлении графа

Очевидно, что наиболее понятный и полезный для человека способ представления графа — изображение графа на плоскости в виде точек и соединяющих их линий — будет совершенно бесполезным, если мы хотим решать с помощью ЭВМ задачи, связанные с графами. Выбор соответствующей структуры данных для представления графов оказывает принципиальное влияние на эффективность алгоритмов.

В теории графов классическим способом представления графа служит матрица инциденций. Для графа $G =< V,\ E >$ — это матрица $I(G)$ с n строками, соответствующими вершинам, $|V| = n$, и с m столбцами, $|E| = m$, соответствующими ребрам. Для ориентированного графа столбец, соответствующий ребру $< x,\ y >$ содержит -1 в строке, соответствующей вершине x, 1 в строке, соответствующей вершине y, и нули во всех остальных строках (петлю $< x,\ x >$ иногда представляют значением 2). В случае неориентированного графа столбец, соответствующий ребру $\{x,\ y\}$, содержит 1 в строках, соответствующих x и y, и нули в остальных строках.

Пример.

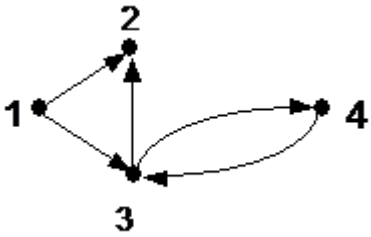


Рис. 3.2

Для ориентированного графа, изображенного на рис. 3.2, матрица инциденций имеет вид:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} < 1,\ 2 > & < 1,\ 3 > & < 3,\ 2 > & < 3,\ 4 > & < 4,\ 3 > \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix} \end{array}$$

С алгоритмической точки зрения матрица инциденций является, вероятно, самым худшим способом представления графа, который только можно представить. Во-первых, он требует m n ячеек памяти. Доступ неудобен. Ответ на элементарные вопросы типа «существует ли ребро $< x,\ y >$?», «к каким вершинам ведут ребра из x?», требует перебора всех столбцов матрицы.

Более удобным способом представление графа является матрица смежности (вершин), определяемая как матрица $B = ||b_{ij}||$ размера $n \times n$, где $b_{ij} = 1$, если существует ребро, ведущее из вершины x в вершину y, $b_{ij} = 0$ в противном случае. Здесь мы подразумеваем, что ребро $\{x,\ y\}$ неориентированного графа идет как от x к y, так и от y к x, так что матрица смежности такого графа всегда является симметричной. Ответ на вопрос типа «$\{x,\ y\} \in E$?» может быть получен за один шаг.

Недостаток такого способа представления графа — $n \times n$ ячеек занятой памяти независимо от числа ребер.

Более экономным в отношении требуемого объема памяти (особенно для неплотных графов $m \ll n \times n$) является метод представления графа с помощью списка пар. Пара (x, y) соответствует ребру $< x,\ y >$, если граф ориентированный, и ребру $\{x,\ y\}$, если граф неориентированный. Очевидно, что объем памяти в этом случае составляет $2m$ ячеек. Неудобство — большое число шагов, порядка m в худшем случае, необходимое для получения множества вершин, к которым ведут ребра из данной вершины.

Для приведенного на рис. 3.2. графа список пар имеет следующий вид:

$$(1,\ 2),\quad (3,\ 2),\quad (4,\ 3),$$
$$(1,\ 3),\quad (3,\ 4).$$

Ситуацию можно значительно улучшить, упорядочив множество пар лексикографически и применяя двоичный поиск, но лучшим решением во многих случаях оказывается структура данных, которая называется списками инцидентности.

Она содержит для каждой вершины $v \in V$ список вершин u, таких что $< v,\ u >\in E$ (или $\{v,\ u\} \in E$ в случае неориентированного графа). Для графа. представленного на рис. 3.2, список инцидентности имеет следующий вид:

Вершина	Список инциденций					
1	1	$\bullet \rightarrow$	2	$\bullet \rightarrow$	3	$\varnothing$
2	2	$\varnothing$				
3	3	$\bullet \rightarrow$	2	$\bullet \rightarrow$	4	$\varnothing$
4	4	$\bullet \rightarrow$	3	$\varnothing$		

Каждый элемент списка инцидентности имеет вид

1	$\bullet \rightarrow$

где

$$\underbrace{1}_{\text{Вершина}} \quad | \qquad \underbrace{\bullet \rightarrow}_{\text{Ссылка на следующий элемент списка}} \qquad |$$

3.3 Поиск в графе

Будем далее рассматривать ориентированные и неориентированные графы без петель и кратных ребер, которые будем называть простыми. Существует много алгоритмов на графах, в основе которых лежит систематический

перебор вершин графа, такой, что каждая вершина просматривается в точности один раз. Поэтому важной задачей является нахождение хороших методов поиска в графе. Вообще говоря, метод поиска «хороший», если:

1) он позволяет легко применить этот метод в алгоритме решения задачи («погрузить» алгоритм решения нашей задачи в этот метод);

2) каждое ребро графа анализируется не более одного раза (или, что существенно не меняет ситуации, число раз, ограниченное константой).

Опишем теперь такой метод поиска в неориентированном простом графе, который стал одним из основных методов проектирования алгоритмов на графах. Этот метод называется *методом поиска в глубину*, по причинам, которые вскоре станут ясными.

3.3.1 Поиск в глубину в графе

Общая идея этого метода состоит в следующем. Мы начинаем поиск с некоторой фиксированной вершины v_0. Затем выбираем произвольную вершину u, смежную с v_0 ($(v_0,\ u) \in E$), и повторяем процесс от u. В общем случае предположим, что мы находимся в вершине v. Если существует новая (еще непросмотренная) вершина u, $(v,\ u) \in E$, то мы рассматриваем эту вершину (она перестает быть новой) и, начиная с нее, продолжаем поиск. Если же не существует ни одной новой вершины, смежной с v, то мы говорим. что вершина v использована, возвращаемся в вершину, из которой мы попали в v, и продолжаем процесс (если $v = v_0$, то поиск закончен). Таким образом, поиск в глубину из вершины v основывается на поиске в глубину из всех новых вершин, смежных с v.

Этот процесс поиска из вершины v для неориентированного простого графа, заданного списком инцидентности СПИСОК (СПИСОК[v] — список вершин, смежных (инцидентных) с вершиной v) легко описать с помощью следующей рекурсивной процедуры:

```
1: procedure ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_G(v) {поиск в глубину из вершины v; переменные НОВЫЙ, ЗАПИСЬ — глобальные}
2: begin
3: рассмотреть v;
4: НОВЫЙ[v] :=ложь;
5: for u ∈ ЗАПИСЬ[v] do
6:   if НОВЫЙ[u] then
7:     ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_G(u);
8: end {вершина v использована}
```

Поиск в глубину в произвольном, необязательно связном, неориентированном простом графе проводится по следующему алгоритму:

```
1: begin
2: for v ∈ V do
3:    НОВЫЙ[v] :=истина; {инициализация}
4: for v ∈ V do
5:    if НОВЫЙ[v] then
6:       ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_G(v);
7: end
```

Покажем теперь, что этот алгоритм просматривает каждую вершину в точности один раз и его сложность порядка $O(n+m)$. Отметим сначала, что вызов ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_$G(v)$ влечет за собой просмотр всех вершин связной компоненты графа, содержащей v (если НОВЫЙ$[u]$ =**истина** для каждой вершины u этой компоненты). Это непосредственно следует из структуры процедуры ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_$G(v)$: после посещения вершины (строка 3) следует вызов процедуры для всех ее новых соседей. Отметим также, что каждая вершина графа просматривается не более одного раза, так как просматриваться может только вершина v, для которой НОВЫЙ$[v]$ = **истина**, сразу же после посещения этой вершины исполняется присваивание НОВЫЙ$[v]$:= **ложь** (строка 4 процедуры).

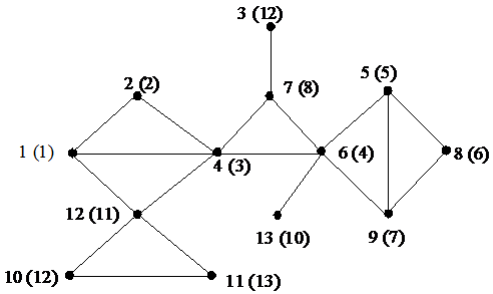


Рис. 3.3

На рис. 3.3. показан граф, вершины которого перенумерованы (номера в скобках) в соответствии с тем порядком, в котором они просматриваются в процессе поиска в глубину (мы отождествляем вершины графа с числами 1, ..., 13 и полагаем, что в списке СПИСОК$[v]$ вершины упорядочены по возрастанию).

Алгоритм начинает поиск поочередно от каждой еще не просмотренной вершины, следовательно, просматриваются все вершины графа (необязательно связного).

Чтобы оценить сложность алгоритма, отметим сначала, что число шагов в обоих циклах (строки 2-5 алгоритма) порядка n, не считая шагов, выполнение которых инициировано вызовом процедуры ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_G. Эта процедура выполняется не более n раз во втором цикле сразу после посещения каждой вершины для каждого из ее новых соседей, итого суммарно $O(n+m)$ раз. Полное число шагов, выполняемое циклом в строке 5 процедуры ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_G, для всех вызовов этой процедуры будет порядка m, где m — число ребер. Это дает общую сложность алгоритма $O(n+m)$.

Отметим, что алгоритм поиска в глубину в графе легко модифицировать так, чтобы он вычислял связные компоненты графа.

В связи с тем, что поиск в глубину играет важную роль в проектировании алгоритмов на графах, представляет интерес нерекурсивная версия процедуры ПОИСК_В_ГЛУБИНУ_В_ГРАФЕ_G. Рекурсия устраняется стандартным способом при помощи стека. Каждая просмотренная вершина помещается в стек и удаляется из стека после ее использования.

Метод поиска в глубину очевидным образом переносится на ориентированные графы.

3.3.2 Поиск в ширину в графе

Теперь рассмотрим несколько иной метод поиска в графе, называемый поиском в ширину. Прежде чем описать его, отметим, что при поиске в глубину чем позднее будет посещена вершина, тем раньше она будет использована - точнее, так будет при допущении, что вторая вершина посещается перед использованием первой. Это прямое следствие того факта, что просмотренные, но еще не использованные вершины накапливаются в стеке. Поиск в ширину основывается, грубо говоря, на замене стека очередью. После такой модификации чем раньше посещается вершина (помещается в очередь), тем раньше она используется (удаляется из очереди). Использование вершины происходит с помощью просмотра всех еще непросмотренных соседей этой вершины. Вся процедура представлена ниже:

```
1: procedure ПОИСК_В_ШИРИНУ_В_ГРАФЕ (v); {поиск в ширину
   в графе с началом в вершине v; переменные НОВЫЙ, СПИСОК —
   глобальные}
2: begin
3: ОЧЕРЕДЬ: = ∅;
4: ОЧЕРЕДЬ ⇐ v;
5: НОВЫЙ[v]: = ложь;
6: while ОЧЕРЕДЬ ≠ ∅ do
7:    begin
8:    p ⇐ ОЧЕРЕДЬ;
9:    посетить p;
10:   for u ∈ ЗАПИСЬ[p] do
11:      if НОВЫЙ[u] then
12:         begin
13:         ОЧЕРЕДЬ ⇐ u;
14:         НОВЫЙ[u]:=ложь;
15:         end
16:   end
17: end {вершина v использована}
```

Способом, аналогичным тому, который был применен для поиска в глубину, можно легко показать, что вызов процедуры ПОИСК_В_ШИРИНУ_В_ГРАФЕ (v) приводит к посещению всех вершин связной компоненты графа, содержащей вершину v, причем каждая вершина просматривается в точности один раз. Вычислительная сложность алгоритма поиска в ширину также имеет порядок $m + n$, так как каждая вершина помещается в очередь и удаляется из очереди в точности один раз, а число итераций цикла 10, очевидно, будет иметь порядок числа ребер графа.

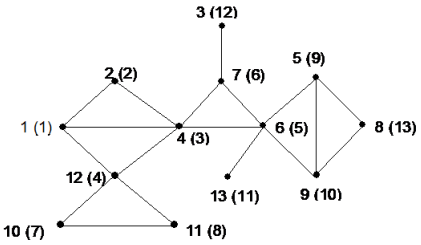


Рис. 3.4

На рис. 3.4 представлен граф, вершины которого занумерованы (в скобках) согласно очередности, в которой они посещаются в процессе поиска в ширину.

Как и в случае поиска в глубину, процедуру ПОИСК_В_ШИРИНУ_В_ГРАФЕ (v) можно использовать без всяких модификаций и тогда, когда список инцидентности СПИСОК[v], $v \in V$, определяет некоторый ориентированный граф. Очевидно, что тогда посещаются только те вершины, до которых существует путь от вершины, с которой мы начинаем поиск.

3.4 Пути и циклы

Определение. *Путем* в ориентированном или неориентированном графе $G =< V,\ E >$ называют последовательность ребер вида $< (v_1,\ v_2),\ (v_2,\ v_3),\ \ldots, (v_{n-1},\ v_n) >= S =< E_1,\ \ldots,\ E_{n-1} >$, где $E_i = (v_i,\ v_{i+1}) \in E$, E_i и E_{i+1} инцидентны одной вершине. Говорят, что этот путь идет из v_1 в v_n и имеет длину $n-1$. Часто такой путь представляют последовательностью вершин $< v_1,\ \ldots,\ v_n >$, $< v_i,\ v_{i+1} >\in E$, лежащих на нем. В вырожденном случае одна вершина обозначает путь длины 0, идущий из этой вершины в нее же.

Путь называется *простым*, если все ребра и все вершины на нем, кроме быть может первой и последней, различны.

Определение. *Цикл* — это простой путь длины не менее 1, который начинается и заканчивается в одной и той же вершине. Заметим, что в неориентированном простом графе длина цикла должна быть не менее 3.

Оба вида поиска в графе — в глубину и в ширину — могут быть использованы для нахождения пути между фиксированными вершинами v и u. Достаточно начать поиск в графе с вершины v и вести его до момента посещения вершины u. Преимуществом поиска в глубину является тот факт, что в момент посещения вершины u стек содержит последовательность вершин, определяющих путь из v в u. Это становится очевидным, если отметить, что каждая вершина, помещаемая в стек, является смежной с верхним элементом стека. Однако недостатком поиска в глубину является то, что полученный таким образом путь в общем случае не будет кратчайшим путем из v в u.

От этого недостатка свободен метод нахождения пути, основанный на поиске в ширину. Модифицируем процедуру ПОИСК_В_ШИРИНУ_В_ГРАФЕ (v), заменяя строки 11—15 на

if НОВЫЙ[u] **then**
begin
ОЧЕРЕДЬ $\Leftarrow u$;
НОВЫЙ[u] := ложь;
ПРЕДЫДУЩИЙ[u] := p;
end

По окончании работы модифицированной таким образом процедуры массив ПРЕДЫДУЩИЙ содержит для каждой просмотренной вершины u вершину ПРЕДЫДУЩИЙ[u], из которой мы попали в u. Отметим, что кратчайший путь из вершины u в вершину v обозначается последовательностью вершин $u = u_1,\ u_2,\ \ldots,\ u_k = v$, где u_{i+1} = ПРЕДЫДУЩИЙ[u_i] для $1 \leq i < k$ и k является первым индексом i для которого $u_i = v$. Действительно, в очереди помещены сначала вершины, находящиеся на расстоянии 0 от v (т.е. сама вершина v), затем поочередно все новые вершины, находящиеся на расстоянии 1 от v, и т.д. Под расстоянием здесь мы понимаем длину кратчайшего пути. Предположим теперь, что мы уже рассмотрели все вершины, находящиеся на расстоянии, не превосходящем r от v, что очередь содержит все вершины, находящиеся на расстоянии r от v, и только эти вершины и что массив ПРЕДЫДУЩИЙ правильно определяет кратчайший путь от каждой, уже просмотренной вершины до вершины v способом, описанным выше. Использовав каждую вершину p, находящуюся в очереди, наша процедура просматривает некоторые новые вершины, и каждая такая новая вершина u находится на расстоянии $r+1$ от v, причем, определяя ПРЕДЫДУЩИЙ[u] := p, мы продлеваем кратчайший путь от p до v до кратчайшего пути от u до v. После использования всех вершин из очереди, находящихся на расстоянии r от v, она (очередь), очевидно, содержит множество вершин, находящихся на расстоянии $r+1$ от v, и легко заметить, что условие индукции выполняется и для расстояния $r+1$.

3.5 Связность

Определение. Пусть граф G — неориентированный. Две вершины a и b называются *связанными*, если существует путь S с начальной вершиной a и конечной вершиной b, $S =< a,\ a_1,\ \ldots,\ a_n,\ b >$. Если S проходит через какую-нибудь вершину a_i более одного раза, то можно, очевидно, удалить его циклический участок и при этом остающиеся ребра будут составлять путь S из a в b. Отсюда следует, что связанные путем вершины связаны и простым путем. Граф называется связным, если любая его пара вершин

связана. Для всякого графа существует такое разбиение множества его вершин

$$V = \bigcup_i V_i$$

на попарно непересекающиеся подмножества вершин V_i, что вершины в каждом V_i связаны, а вершины из различных V_i не связаны.

Граф H называется *частью графа* G, если множество его вершин $V(H)$ содержится во множестве вершин $V(G)$ графа G и все ребра H являются ребрами G. Для любой части H графа G существует единственная дополнительная часть (дополнение) $\overline{H}$, состоящее из всех ребер графа G, которые не вошли в H, и инцидентных им вершин. Особенно важным типом частей являются *подграфы*.

Пусть V' — подмножество вершин графа $G =< V,\ E >$, $V' \in V$. *Подграф* $G(V',\ E')$, определяемый множеством V', есть такая часть графа G, множеством вершин которой является V', а ребрами — все ребра из G, оба конца которых лежат в V':

$$G(V',\ E') = G(V',\ E' = \{(u,\ v)|((u,\ v) \in E) \wedge (u,\ v \in V')\}).$$

Тогда в соответствии с разбиением $V = \bigcup_i V_i$ мы получаем прямое разложение

$$G = \bigcup_i G(V_i,\ E_i)$$

графа G на непересекающиеся связные подграфы $G(V_i)$. Эти подграфы называются *связными компонентами* (или *компонентами связности*) графа G.

Теорема 3.2. *Если в конечном неориентированном простом графе G ровно две вершины a_0 и b_0 имеют нечетную локальную степень, то они связаны.*

Доказательство. По теореме 3.1, каждый конечный граф имеет четное число вершин нечетной степени. Так как это условие выполняется и для той компоненты G, которой принадлежит a_0, то b_0 принадлежит той же компоненте связности. □

Теорема 3.3. *Если неориентированный простой граф G имеет n вершин и k связных компонент, то максимальное число ребер в G равно*

$$N(n,\ k) = \frac{1}{2}(n-k)(n-k+1).$$

Доказательство. Пусть в графе G связная компонента G_i имеет n_i вершин. Тогда максимальное число ребер в G равно $N = \frac{1}{2}\sum_{i=1}^{k} n_i(n_i - 1)$. Это число достигается, когда каждый из графов G_i полный и имеет n_i вершин. Допустим, что среди графов G_i найдутся хотя бы два, которые имеют более одной вершины, например $n_2 \geq n_1 > 1$. Построим вместо G другой граф G' с тем же числом вершин и компонент, заменяя G_1 и G_2 полными графами G_1' и G_2' соответственно с $n_1 - 1$ и $n_2 + 1$ вершинами. Легко видеть, что это увеличит число ребер. Таким образом максимальное число ребер должен иметь граф, состоящий из $k - 1$ изолированных вершин и одного полного графа с $n - k + 1$ вершинами. □

Из теоремы 3.3 следует для случая k = 2 следующее утверждение.

Теорема 3.4. *Простой неориентированный граф с n вершинами и с числом ребер, большим, чем*

$$N(n,\ 2) = \frac{1}{2}(n-1)(n-2)$$

связен.

3.6 Деревья

Определение. Связный неориентированный граф называется *деревом*, если он не имеет циклов.

В частности, дерево не имеет петель и кратных ребер. Граф без циклов есть граф, связные компоненты которого являются деревьями; иногда такой граф называется *лесом*. Любая часть такого графа также будет графом без циклов.

Теорема 3.5. *В дереве любые две вершины связаны единственным простым путем.*

Доказательство. Любой путь в графе без циклов является простым. Если бы было два связывающих вершины простых пути, то был бы и цикл в дереве, что невозможно. □

Наглядное представление для произвольного дерева $T =< V,\ E >$ можно получить при помощи следующей конструкции.

Выберем произвольную вершину a_0 и будем рассматривать ее как *корень дерева* или вершину нулевого уровня.

$$U_0 = \{a_0\}.$$

От a_0 проведем все ребра к вершинам, находящимся на расстоянии 1 от вершины a_0. Вершины, смежные с a_0 составят множество вершин первого уровня:

$$U_1 = \{a_i^{(1)} | \{a_0,\ a_i^{(1)}\} \in E\}.$$

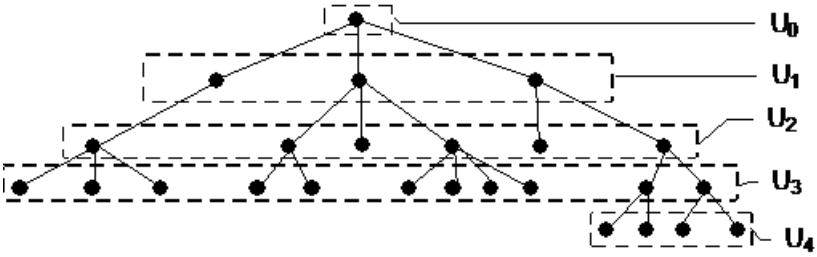
От этих вершин первого уровня проведем все ребра к смежным с ними вершинам, находящимся на расстоянии 2 от a_0, за исключением ребер, инцидентных вершинам первого уровня. Получим множество вершин второго уровня

$$U_2 = \{a_j^{(2)} | \{a_i^{(1)},\ a_j^{(2)}\} \in E,\ a_i^{(1)} \in U_1, a_j^{(2)} \notin U_1\}$$

Из вершины $a_i^{(n)}$ находящейся на расстоянии n от a_0, выходит одно ребро к единственной предшествующей вершине $a_k^{(n-1)}$, а также некоторое семейство ребер к вершинам $a_j^{(n+1)}$, находящимся на расстоянии $n+1$.

$$U_n = \{a^{(n)} | \{a^{(n-1)},\ a^{(n)}\} \in E,\ a^{(n)} \notin U_{n-1}\}.$$

Ни для какой из этих вершин $a^{(n)}$ не может быть ребер, соединяющих ее с вершинами с тем же или меньшим расстоянием, кроме $\{a^{(n-1)},\ a^{(n)}\}$. Таким образом, дерево может быть представлено в следующей форме:



Будем называть вершину a дерева *концевой*, если $\rho(a) = 1$.

Будем называть ребро *концевым*, если хотя бы одна инцидентная ему вершина является концевой.

Утверждение 3.6. *Любое нетривиальное конечное дерево имеет хотя бы две концевые вершины и хотя бы одно концевое ребро.*

Доказательство совершенно просто может быть проведено, например, индукцией по числу вершин.

Утверждение 3.7. *Каждое дерево с n вершинами имеет $n-1$ ребро.*

Доказательство. Доказательство легко проводится индукцией по числу вершин. Для $n = 1$ утверждение, очевидно, справедливо. Пусть $n > 1$. Тогда в дереве существует концевая вершина v. Удаляя из дерева v и ребро $(u,\ v)$, инцидентное v, получим дерево с $n-1$ вершиной, которое в силу индуктивного предположения имеет $n-2$ ребра. Следовательно, первоначальное дерево имеет $n-2+1 = n-1$ ребро. □

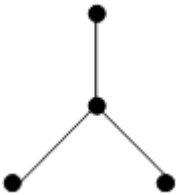
Обратимся теперь к вопросу о подсчете числа деревьев, которые могут быть построены на заданном множестве вершин. Подчеркнем, что речь идет не о подсчете числа различных попарно неизоморфных деревьев, а именно о числе деревьев, графы которых различны, то есть различаются хотя бы одним ребром (множество вершин фиксировано).

Пример. Пусть количество вершин n равно 4. Тогда на этом множестве вершин могут быть построены следующие различные деревья.

Первый класс таких деревьев составляют деревья следующего вида:

Концевые вершины для деревьев такого класса могут быть выбраны $\binom{4}{2}$ способами, при каждом таком выборе порядок внутренних вершин может быть выбран двумя способами. Получаем 12 различных деревьев, составляющих указанный класс.

Второй класс деревьев имеет вид:



Центральная вершина может быть выбрана 4 способами, что дает 4 различных дерева в этом классе.

Других деревьев с 4 вершинами, отличных от указанных в первом и втором классах, нет. Таким образом, существует 16 деревьев, имеющих 4 фиксированные вершины.

Обратимся теперь к общему результату.

Теорема 3.8. *Число различных деревьев, которые можно построить на заданном множестве V, содержащем n вершин, равно*

$$t_n = n^{n-2}$$

Доказательство. Обозначим элементы данного множества V, расположенные в некотором фиксированном порядке, числами

$$V = \{1,\ 2,\ \ldots,\ n\} \tag{3.1}$$

Для любого дерева T, определенного на V, введем некоторый символ, характеризующий его однозначно. В T существуют концевые вершины. Обозначим через b_1 первую концевую вершину в последовательности (3.1), а через $e_1 = \{a_1,\ b_1\}$ — соответствующее концевое ребро. Удалив из T ребро e_1 и вершину b_1, мы получим новое дерево T_1. Для T_1 найдется первая в списке (3.1) концевая вершина b_2 с ребром $e_2 = \{a_2,\ b_2\}$. Эта редукция повторяется до тех пор, пока после удаления ребра $e_{n-2} = (a_{n-2},\ b_{n-2})$ не останется единственное ребро $e_{n-1} = (a_{n-1},\ b_{n-1})$, соединяющее две оставшиеся вершины.

Тогда список

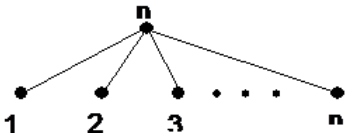
$$\sigma(T) = [a_1,\ a_2,\ \ldots,\ a_{n-2}] \tag{3.2}$$

однозначно определяются деревом T и двум различным деревьям T и T', очевидно, соответствуют разные символы такого вида. Каждая вершина a_i появляется в (3.2) $\rho(a_i) - 1$ раз.

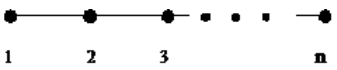
Для ясности приведем несколько примеров деревьев и соответствующих им последовательностей $\sigma(T)$.

Примеры.

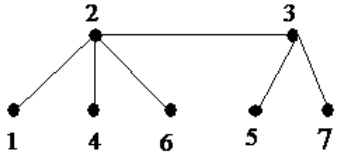
1. $\sigma(T) = [\underbrace{n,\ n,\ \ldots,\ n}_{n-2}]$



2. $\sigma(T) = [2,\ 3,\ \ldots,\ n-1]$



3. $\sigma(T) = [2,\ 2,\ 3,\ 2,\ 3]$



Наоборот, каждая последовательность (3.2) определяет дерево T с помощью обратного построения. Если дана последовательность (3.2), то находится первая вершина b в списке (3.1), которая не содержится в (3.2). Это определяет ребро $e_1 = \{a_1,\ b_1\}$. Далее удаляем вершины a_1 из последовательности (3.2) и b_1 из списка (3.1) и продолжаем построение для оставшихся элементов.

Получающийся в результате построения граф является деревом, что может быть установлено, например, по индукции. После удаления e_1 последовательность (3.2) будет содержать $n-3$ элемента. Если она соответствует дереву T_1, то граф T, получаемый из него добавлением e_1, также есть дерево, так как вершина b_1 не принадлежит T_1.

В (3.2) каждая вершина может принимать все n возможных значений. Все они соответствуют различным деревьям, откуда и получается формула $t_n = n^{n-2}$. □

3.6.1 Остовное дерево (каркас)

Определение. Для связного неориентированного графа $G =< V,\ E >$ без петель каждое дерево $< V,\ T >$, содержащее все вершины графа G, где $T \subseteq E$, будем называть *остовным деревом (каркасом)* графа G.

Напомним, что длина пути есть количество составляющих его ребер.

Процедуры поиска в глубину и в ширину можно простым способом использовать для нахождения остовных деревьев. В обоих случаях достижение новой вершины u из вершины v вызывает включение в дерево ребра $(v,\ u)$.

Приведем алгоритм построения остовного дерева методом поиска в глубину.

Алгоритм построения остовного дерева методом поиска в глубину.

Входные данные: связный граф $G =< V,\ E >$, заданный списками инцидентности ЗАПИСЬ$[v]$, $v \in V$.

Результат: каркас $< V,\ T >$ графа G.

```
1: procedure КАРКАС_ГЛУБИНА(v) {поиск в глубину из вершины v,
   соединенный с нахождением ребра дерева; переменные НОВЫЙ, ЗА-
   ПИСЬ, T — глобальные}
2: begin
3: НОВЫЙ[v]: = ложь;
4: for u ∈ЗАПИСЬ[v] do
5:    if НОВЫЙ[u] then
6:       (u, v) — новая ветвь;
7:    begin
8:    T := T ∪ (v, u);
9:    КАРКАС_ГЛУБИНА(u);
10:   end
11: end {вершина v использована};
12: begin {главная программа}
13: for u ∈ V do
14:    НОВЫЙ [u] := истина; {инициализация}
15: T := ∅; {T — множество найденных к этому времени рёбер каркаса}
16: КАРКАС_ГЛУБИНА(r) {r — произвольная вершина графа}
17: end
```

Для доказательства того, что приведенный алгоритм правильно строит каркас произвольного связного графа, достаточно отметить следующие три факта. Во-первых, в момент добавления к множеству T нового ребра $(v,\ u)$ в строке 8 в $< V,\ T >$ существует путь из r в v (этот факт легко доказывается по индукции). Таким образом, алгоритм строит связный граф. Во-вторых, каждое новое ребро $(v,\ u)$, добавляемое ко множеству T, соединяет уже рассмотренную вершину v (т.е. НОВЫЙ $[v]$ = ложь) с новой вершиной u. Отсюда следует, что построенный граф $< V,\ T >$ не содержит циклов. Действительно, последнее ребро, «замыкающее» цикл, должно было бы соединить две уже рассмотренные вершины. И, наконец, в-третьих, из свойства поиска в глубину следует, что процедура КАРКАС_ГЛУБИНА просматривает все вершины связного графа. Следовательно, граф $< V,\ T >$, построенный нашим алгоритмом, есть остовное дерево графа G. Вычислительная сложность алгоритма есть, очевидно, $O(n+m)$, т.е. того же порядка, что и сложность поиска в глубину.

Аналогично выглядит процедура построения остовного дерева методом поиска в ширину.

Алгоритм построения остовного дерева методом поиска в ширину.

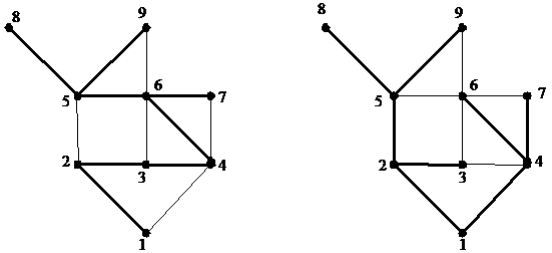
Входные данные: связный граф $G =< V,\ E >$, заданный списками инцидентности ЗАПИСЬ$[v]$, $v \in V$.

Результат: каркас $< V,\ T >$ графа G.

```
1:  begin
2:  for u ∈ V do
3:    НОВЫЙ [u] := истина; {инициализация}
4:  T := ∅; {T — множество найденных к этому времени ребер каркаса}
5:  ОЧЕРЕДЬ := ∅;
6:  ОЧЕРЕДЬ ⇐ r;
7:  НОВЫЙ [r] := ложь; {r — корень каркаса}
8:  while ОЧЕРЕДЬ ≠ ∅ do
9:    begin
10:   p ⇐ ОЧЕРЕДЬ;
11:   for u ∈ЗАПИСЬ[v] do
12:     if НОВЫЙ[u] then
13:       begin
14:       ОЧЕРЕДЬ ⇐ u;
15:       НОВЫЙ[u] := ложь;
16:       T := T ∪ (v, u);
17:       end
18:   end
19: end
```

Рассуждениями, аналогичными проведенным для алгоритма КАРКАС_ГЛУБИНА, можно показать, что данный алгоритм корректно строит остовное дерево для произвольного связного графа за $O(n + m)$ шагов.

На следующем рисунке дан пример остовного дерева для графа, построенного методом поиска в ширину (слева) и в глубину (справа).



Каждое остовное дерево, построенное с помощью метода поиска в глубину, имеет любопытное свойство, которое сейчас будет описано.

Вершину r, из которой начинается поиск, назовем *корнем остовного дерева*. Для двух различных вершин v и u дерева $< V,\ T >$ будем говорить, что u является потомком вершины v, если v лежит на пути (в дереве $< V,\ T >$) из вершины u в вершину v.

Теорема 3.9. *Пусть $< V,\ T >$ — остовное дерево связного неориентированного графа $G =< V,\ E >$, построенное с помощью алгоритма КАРКАС_ГЛУБИНА, и пусть $(u,\ v) \in E$. Тогда либо u — потомок v, либо v — потомок u.*

Доказательство. Предположим без ограничения общности, что вершина v будет просмотрена раньше, чем u. Рассмотрим процесс поиска в глубину, начиная с вершины v. Очевидно, что по окончании его должно быть НОВЫЙ$[u]$ = **ложь**, ибо $(v,\ u)$ — ребро. Но это означает, что ребра, добавленные во множество T в течение этого процесса, содержат путь из v в u, откуда следует, что v лежит на пути из u в корень, поскольку в дереве существует в точности один путь из произвольной вершины к корню. □

Рассуждения, проведенные в предыдущем разделе, непосредственно приводят к следующей теореме.

Теорема 3.10. *Пусть $< V,\ T >$ — остовное дерево связного неориентированного графа $G =< V,\ E >$, построенное с помощью алгоритма поиска в ширину. Тогда путь в $< V,\ T >$ из произвольной вершины v до корня r является кратчайшим путем из v в r в графе G.*

Далее особо обсуждается более общая задача отыскания кратчайших путей в графе, ребрам которого приписаны «длины» (веса), не обязательно равные единице.

3.7 Эйлеровы пути и циклы

Задача о кенигсбергских мостах послужила началом теории графов. План расположения семи мостов в Кёнигсберге (ныне г. Калининград) приведен на рис. 3.5.

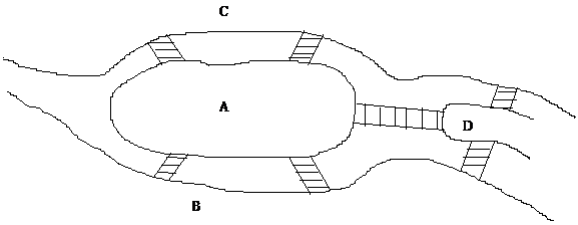
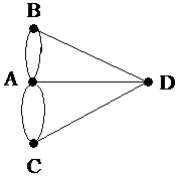


Рис. 3.5

Задача состоит в том, чтобы пройти каждый мост по одному разу и вернуться в исходную точку C. Так как существенны только переходы через мосты, план города можно свести к изображению графа, в котором ребра соответствуют мостам, а вершины — разделенным частям города.



Очевидно, не существует циклических обходов, проходящих по всем ребрам по одному разу.

Развлечения, в которых требуется обрисовать некоторую фигуру, не прерывая и не повторяя линии, являются, по-видимому, очень давними. Считается, что фигура, называемая саблями Магомета, имеет арабское происхождение.

Эйлер обратился к общей задаче, касающейся графов: в каких случаях в конечном графе можно найти такой цикл, в котором каждое ребро участвовало бы один раз?

Определение. *Эйлеровым путем* в графе $G =< V,\ E >$ называется произвольный путь, проходящий через каждое ребро графа в точности один раз, т.е. путь $S =< v_1,\ \ldots,\ v_{m+1}$ такой, что каждое ребро $e \in E$ появляется в последовательности $v_1,\ \ldots,\ v_{m+1}$ в точности один раз как $e = \{v_i,\ v_{i+1}\}$ для некоторого i. Если $v_1 = v_{m+1}$, то такой путь называется *эйлеровым циклом.*

Теорема 3.11. *Эйлеров путь в неориентированном простом графе существует тогда и только тогда, когда граф связный и содержит не более, чем две вершины нечетной степени.*

Если в связном графе нет вершин нечетной степени, то каждый эйлеров путь является циклом, так как концы эйлерова пути, не являющегося циклом, всегда вершины нечетной степени. Предположим, что u и v — единственные вершины нечетной степени в связном графе $G =< V,\ T >$ и образуем граф G' добавлением дополнительной вершины t и ребер $\{u,\ v\}$ и $\{v,\ t\}$ (или просто добавлением ребра $\{u,\ v\}$, если $\{u,\ v\} \notin E$). Тогда G' — связный граф без вершин нечетной степени, а эйлеровы пути в G будут в очевидном взаимно однозначном соответствии с эйлеровыми циклами в G'. В силу этого мы будем заниматься только эйлеровыми циклами и переформулируем теорему.

Теорема 3.12. *Конечный граф $G =< V,\ E >$ содержит эйлеров цикл тогда и только тогда, когда*

1. *G — связен.*

2. *Все степени его вершин четные.*

Доказательство. Условие 1, очевидно, необходимо. Далее, каждый раз, когда эйлеров цикл проходит через какую-то вершину, он должен войти в нее по одному ребру и выйти по другому; поэтому условие 2 также необходимо.

Предположим теперь, что эти два условия выполнены. Начнем путь в произвольной вершине a графа G и будем продолжать его, насколько возможно, всё время через новые рёбра. Так как степени всех вершин четны, этот процесс может закончится только в вершине a.

Если содержит не все рёбра графа G, то удалим из G часть , состоящую из ребер этого цикла.

Графы и G имеют четные степени вершин, то же будет справедливо и для остающегося графа $\overline{P}$.

Так как граф G связен, в должна найтись вершина b, инцидентная также ребрам из $\overline{P}$. Из b построим новый путь P', содержащий ребра только из $\overline{P}$. Снова такой путь может закончится только при возвращении в b. Но тогда из и P' составим новый путь P_1

$$P_1 = P(a,\ b) \cup P' \cup P(b,\ a),$$

который возвращается в и содержит больше ребер, чем .

Если P_1 не является эйлеровым циклом, то это построение повторяется. Когда процесс закончится, эйлеров цикл будет построен. □

Представим изложенный процесс построения эйлерова цикла в виде схемы алгоритма на «псевдоалгоритмическом» языке.

3.7.1 Алгоритм построения эйлерова цикла

Граф хранится в виде списка инциденций.

СПИСОК[v] — список вершин графа, смежных с вершиной v.

Метод

Начинаем строить путь с началом в v, причем вершины этого пути помещаются в стек ПУТЬ, а рёбра удаляются из графа.

Это продолжается до тех пор, пока путь можно удлинить, включив в него новую вершину, т.е. СПИСОК [v] $\neq \varnothing$. Это случится только по достижению вершины v. Тем самым из графа удален цикл, а вершины этого цикла находятся в стеке ПУТЬ. Вершина v переносится теперь из списка ПУТЬ в стек ЦИКЛ. Процесс повторяется.

Алгоритм

```
1:  begin
2:  СТЕК: ПУТЬ := ∅;
3:  СТЕК: ЦИКЛ := ∅;
4:  v := произвольная вершина в графе;
5:  ПУТЬ := v; {Инициализация}
6:  while ПУТЬ ≠ ∅ do
7:    begin
8:    v := ПУТЬ;
9:    if СПИСОК[v] ≠ ∅ then
10:     begin
11:     u := ПЕРВАЯ ВЕРШИНА СПИСКА СПИСОК [v];
12:     ПУТЬ := u;
13:     СПИСОК[v] := СПИСОК[v] \ {u};
```

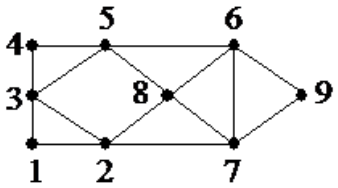
```
14:      СПИСОК[u] := СПИСОК[u] \ {v};
15:      v := u
16:      end
17:    else
18:      begin
19:      v :=ПУТЬ;
20:      ЦИКЛ:= v
21:      end
22:    end
23: end
```

Оценим вычислительную сложность алгоритма. Для этого отметим, что каждое повторение главного цикла либо помещает вершину в стек ПУТЬ и удаляет ребро из графа, либо переносит вершину из стека ПУТЬ в стек ЦИКЛ. Таким образом, число итераций этого цикла — $O(m)$, где m — число ребер. В свою очередь число шагов в каждом повторе ограничено константой. Общая сложность алгоритма есть $O(m)$.

Приведем пример построения эйлерова цикла с помощью описанного алгоритма.

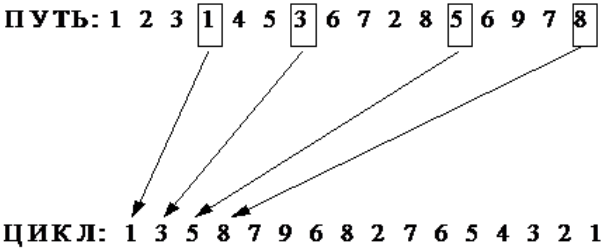
Пример

Пусть граф имеет следующий вид:



Построим списки смежности вершин для указанного графа:
СПИСОК [1]: 2, 3;
СПИСОК [2]: 1, 3, 7, 8;
СПИСОК [3]: 1, 2, 4, 5;
СПИСОК [4]: 3, 5;
СПИСОК [5]: 3, 4, 6, 8;
СПИСОК [6]: 5, 7, 8, 9;
СПИСОК [7]: 2, 6, 8, 9;
СПИСОК [8]: 2, 5, 6, 7;
СПИСОК [9]: 6,7.

Во время работы алгоритма стеки ПУТЬ и ЦИКЛ имеют следующий вид:



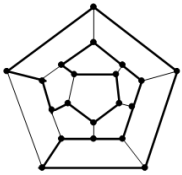
Прямоугольниками в стеке ПУТЬ обрамлены вершины v, при достижении которых СПИСОК $[v] = \varnothing$. Такие вершины немедленно помещаются в стек ЦИКЛ. По исчерпании списка всех вершин (в нашем примере при достижении последней вершины 8) вершины из стека ПУТЬ последовательно выталкиваются в стек ЦИКЛ.

3.8 Гамильтоновы пути и циклы

Рассмотрим теперь задачу предыдущего раздела с той лишь разницей, что на этот раз нас будут интересовать пути, проходящие в точности один раз через каждую вершину (а не каждое ребро) данного графа. Эта небольшая, как может показаться, модификация приводит к значительному изменению характера проблемы.

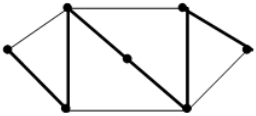
Определение. Простой путь (цикл) называется *гамильтоновым путем (циклом)*, если он проходит через каждую вершину графа.

Слово «гамильтонов» в этих определениях связано с именем известного ирландского математика У. Гамильтона, который в 1859 г. предложил игру «Кругосветное путешествие». В этой игре мир представлен додекаэдром, каждой вершине которого приписано название одного из городов мира. Требуется, переходя из одного города в другой по ребрам додекаэдра, посетить каждый город в точности один раз и вернуться в исходный город.

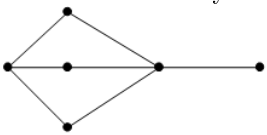


Более толстыми линиями на графе додекаэдра выделен гамильтонов путь.

Укажем аналогичным выделением гамильтонов путь для следующего графа:



С другой стороны, легко видеть, что в следующем графе не существует гамильтоновых путей:



В отличие от эйлеровых путей не известно ни одного простого необходимого и достаточного условия для существования гамильтоновых путей и циклов, и это несмотря на то, что задача — одна из центральных в теории графов. Не известен так же алгоритм, который проверял бы существование гамильтонова пути в произвольном графе, используя число шагов, ограниченное многочленом от переменной n (числа вершин в графе). Имеются веские основания, но нет математических доказательств того, чтобы предполагать, что такое положение вызвано не нашим незнанием, а скорее тем фактом, что такого алгоритма не существует. Проблема существования гамильтонова пути принадлежит к классу так называемых NP-полных задач. Это широкий класс задач, включающий задачи из теории графов, логики, теории чисел, дискретной оптимизации, ни для одной из которых не известен полиномиальный алгоритм (то есть алгоритм с числом шагов, ограниченным полиномом от размерности задачи).

В применениях графов к играм вершины соответствуют различным позициям. Существование гамильтонова цикла равносильно существованию циклической последовательности ходов, содержащей каждую позицию по одному разу. (Пример — задача о шахматном коне: можно ли, начиная с произвольного поля доски, ходить конем так, чтобы пройти через все поля и вернуться в исходное.)

Выведем некоторые условия, при которых можно утверждать, что гамильтонов цикл существует. Эти рассмотрения тесно связаны со свойствами максимальных простых путей.

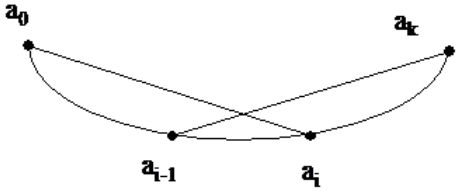
Пусть $S =< a_0,\ a_1,\ \ldots,\ a_k >$ — некоторый простой путь длины k в графе $G =< V,\ E >$.

Определим подграф $G_0 =< \{a_0,\ \ldots,\ a_k\}, E' >=< S,\ E' >$.

Будем говорить, что S имеет *тип цикла*, если подграф $G_0 =< S,\ E' >$, $E' \subseteq E : (\{u,\ v\} \in E' \Leftrightarrow u,\ v \in S,\ \{u,\ v\} \in E)$ имеет гамильтонов цикл.

Отсюда, в частности, следует, что S является гамильтоновым путем в G_0.

Пусть $(a_0,\ a_i) \in E'$ — некоторое ребро в графе G_0. Если существует также ребро $(a_k,\ a_{i-1})$, то в G_0 будет гамильтонов цикл, а именно $(a_0,\ a_i) \cup S(a_i,\ a_k) \cup (a_k,\ a_{i-1}) \cup S(a_{i-1},\ a_0)$.



Если, однако, $\rho_0(a_k) > k - \rho_0(a_0)$, где $\rho_0(a_0)$ и $\rho_0(a_k)$ степени вершин a_0 и a_k в графе G_0, то ясно, что хотя бы для одного ребра $(a_0,\ a_i)$ должно существовать соответствующее ребро $(a_k,\ a_{i-1})$. Поэтому если $\rho_0(a_0) + \rho_0(a_k) \geq k + 1$, то простой путь S будет иметь тип цикла.

Будем говорить, что простой путь — *полный*, если его нельзя продолжить при помощи добавления ребер к какому-нибудь из концов. Тогда все ребра от a_0 и от a_k должны идти к вершинам графа G_0, так что $\rho(a_0) = \rho_0(a_0),\ \rho(a_k) = \rho_0(a_k)$.

Это дает следующую теорему.

Теорема 3.13. *Полный простой путь длины k имеет тип цикла, если*

$$\rho(a_0) + \rho(a_k) \geq k + 1.$$

Теорема 3.14. *Максимальный простой путь в связном графе может иметь тип цикла только тогда, когда граф имеет гамильтонов цикл.*

Доказательство. 1). Если G имеет гамильтонов цикл, то максимальный простой путь длины $n - 1$ имеет тип цикла.

2). Если подграф G_0, соответствующий максимальному простому пути имеет гамильтонов цикл, но G_0 не составляет всего графа, то из-за связности G существует некоторое ребро $(a_i,\ b)$, в котором b не принадлежит G_0. Это, однако, невозможно так как тогда нашелся бы простой путь, который был бы длиннее данного простого пути S. □

Из двух доказанных утверждений следует

Теорема 3.15. *В связном графе либо имеется гамильтонов цикл, либо длина его максимальных простых путей удовлетворяет неравенству*

$$k \geq \rho(a_0) + \rho(a_k).$$

Из последнего условия вытекает, что $k \geq \min(\rho(a_0) + \rho(a_k))$ для всех пар вершин a_0 и a_k, причем можно даже ограничиться теми парами, для которых нет ребра $(a_0,\ a_i)$ Отсюда следует

Теорема 3.16. *В графе без гамильтоновых циклов длина его максимальных простых путей k удовлетворяет неравенству $k \geq \rho_1 + \rho_2$, где ρ_1 и ρ_2 — две наименьшие степени вершин.*

В теореме 3.16 можно не предполагать, что граф связен.

Отметим еще, что в графе с n вершинами максимальный простой путь имеет длину, не превышающую $n - 1$.

Как частный случай приведенных результатов получается следующая теорема.

Теорема 3.17. *Если в графе G с n вершинами для любой пары вершин a_0 и a_k имеет место соотношение $\rho(a_0) + \rho(a_k) \geq n - 1$, то граф G имеет гамильтонов путь.*

Если $\rho(a_0) + \rho(a_k) \geq n$, то G имеет гамильтонов цикл.

Отсюда, в частности, следует результат Дирака о том, что граф с n вершинами имеет гамильтонов цикл, если для каждой его вершины $\rho(a) \geq \frac{1}{2}n$.

Определение. Будем называть неориентированный простой граф *k-связным*, если наименьшее число вершин, удаление которых приводит к несвязному или одновершинному графу, равно k.

Случаи, когда $k = 2$ или $k = 3$ в теории графов имеют особую роль. Такие графы фигурируют во многих теоретических и прикладных вопросах, в частности ряд задач достаточно уметь решать для 2-связных компонент; кроме того, при $k = 3$ и, особенно при $k = 2$ удается дать достаточно обозримое описание соответствующих графов.

Рассмотрим сначала некоторые простые свойства 2-связных графов, вытекающие непосредственно из определения:

1. Степени вершин двусвязного графа больше единицы.

2. Если графы G_1 и G_2 2-связны и имеют не менее двух общих вершин, то граф $G_1 \cup G_2$ также 2-связен.

3. Назовем *точкой сочленения графа* такую его вершину, удаление которой приводит к графу с большим числом компонент связности; если вершина v графа не является точкой сочленения графа, то любые две его вершины соединены путем, не содержащим v; в частности, в 2-связном графе для любых трех несовпадающих вершин a, b, v имеется путь из вершины a в вершину b, не проходящий через вершину v.

Будем в дальнейшем графы, имеющие гамильтонов цикл, называть *гамильтоновыми*. Гамильтонов граф должен быть двусвязным. Однако пример графа, представленный на следующем рисунке, показывает, что для гамильтоновости графа двусвязности недостаточно.

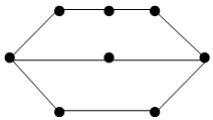


Рис. 3.6

Определение. Любой граф, содержащий две вершины степени 3, соединенных тремя простыми попарно непересекающимися путями длины не менее двух, называется *тэта-графом*. Пример тэта-графа приведен на рис. 3.6.

Далее будет доказана теорема о том, что каждый негамильтонов двусвязный граф содержит тэта-подграф. Но прежде чем переходить к доказательству этого результата, остановимся на необходимом для него утверждении относительно двусвязных графов, представляющем и самостоятельный интерес.

Теорема 3.18. *Пусть* $G =< V,\ E >$ *— связный граф и количество его вершин* $|V| > 2$. *Тогда следующие утверждения эквивалентны:*

1. *Граф 2-связен.*

2. *Любые две вершины графа принадлежат простому циклу.*

3. *Любая вершина и любое ребро принадлежат простому циклу.*

4. *Любые два ребра принадлежат простому циклу.*

5. *Для любых двух вершин a и b и любого ребра e существует простой $(a,\ b)$-путь (путь с начальной вершиной a и конечной вершиной b), содержащий e;*

6. *Для любых трех вершин a, b, c существует простой $(a,\ b)$-путь, проходящий через .*

Доказательство. $1 \Rightarrow 2$. Пусть a и b — две вершины графа G. Рассмотрим множество всех простых циклов графа G, содержащих вершину a. Обозначим через U множество всех вершин, входящих в эти циклы. Ясно, что $U \neq \varnothing$. Действительно, простой цикл, содержащий a, можно получить, объединив два ребра $(a,\ x)$ и $(a,\ y)$ $(x \neq y)$ и простой $(x,\ y)$-путь, не проходящий через a (существующий согласно свойству 3 двусвязных графов). Предположим, что $b \neq U$, и положим $\overline{U} = V \setminus U$. Поскольку граф G связен, то в нем найдется такое ребро $(x,\ t)$, что $x \in U,\ t \in \overline{U}$ (рис. 3.7). Пусть S — простой цикл, содержащий a и x. Так как G-двусвязный граф, то в нем имеется простой $(a,\ t)$-путь P, не содержащий x. Пусть v — первая, считая от t, вершина, входящая в S, т.е. $(t,\ v)$-подпуть пути P не имеет с S общих вершин, отличных от v. Теперь легко построить простой цикл, содержащий v и t. Он получается объединением $(v,\ z)$- пути, проходящего через a и являющегося частью S, с ребром $(z,\ t)$ и $(t,\ v)$-частью пути P (на рис. 3.7 этот цикл показан пунктирной линией). Следовательно, $t \in U$, но это противоречит выбору ребра $(x,\ t)$. Таким образом, $\overline{U} = \varnothing$, т.е. a и b принадлежат одному общему простому циклу.

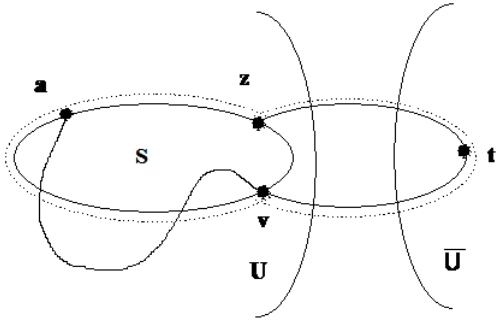


Рис. 3.7

$2 \Rightarrow 3$. Пусть a — вершина и $(x,\ t)$ — ребро графа. По условию G содержит цикл S, проходящий через вершины a и x. Не теряя общности будем считать, что ребро $(x,\ t) \notin S$. Если при этом окажется, что S проходит через вершину t, то требуемый цикл строится очевидным образом. Пусть S не проходит через вершину t. Тогда рассмотрим простой цикл, проходящий через вершины t и a. Такой цикл, по условию, существует. Частью этого цикла является простой путь P, соединяющий t с некоторой вершиной $v \in S$. Путь P можно выбрать так, чтобы пути P и S пересекались только в вершине v. Искомый цикл строится точно так же, как в предыдущем пункте.

$3 \Rightarrow 4$. Пусть $(a,\ b)$ и $(t,\ z)$ — два ребра графа G. По условию G имеет простые циклы S и S', первый из которых содержит ребро $(a,\ b)$ и вершину z, а второй — $(a,\ b)$ и t. Далее искомый цикл строится точно так же, как в предыдущих пунктах.

$4 \Rightarrow 5$. Пусть a и b — вершины графа G, и $(t,\ z)$ — его ребро. Будучи связным, граф G содержит простой путь $P = (a,\ x,\ \ldots,\ b)$. Согласно утверждению 4, в графе G есть простой цикл S, содержащий ребра $(a,\ x)$ и $(t,\ z)$. Легко видеть, что в объединении $S \cup P$ имеется требуемый путь.

$5 \Rightarrow 6$. Пусть a, b, c — вершины графа G, $(c,\ d)$ — его ребро. По условию в графе имеется простой $(a,\ b)$ — путь, проходящий через $(c,\ d)$, и следовательно, содержащий c.

$6 \Rightarrow 1$. Пусть v — вершина графа G. Покажем, что граф G с удаленной вершиной v (граф $G \setminus \{v\}$) — связен, т.е. любая пара a, b его вершин соединена путем. Действительно, согласно утверждению 6 в графе G имеется простой $(v,\ b)$-путь, проходящий через вершину a. Этот путь содержит $(a,\ b)$-подпуть, который, очевидно, не проходит через v и, следовательно, является $(a,\ b)$-путем и в графе $G \setminus \{v\}$. □

Теорема 3.19. *Каждый негамильтонов двусвязный граф содержит тэта-подграф.*

Доказательство. Пусть $G =< V,\ E >$ — негамильтонов двусвязный граф и C — простой цикл максимальной длины в этом графе. По условию теоремы множество S вершин графа G, не принадлежащих циклу S, непусто. Как подтверждает прямая проверка, среди двусвязных графов, порядок которых меньше пяти, нет негамильтоновых, поэтому $|V| \geq 5$. Из двусвязности графа и теоремы 3.18 следует, что количество вершин в цикле C не менее четырех.

Пусть $(x,\ v)$ — такое ребро графа G, что $x \in C$, $v \in S$, a и b — вершины цикла C, смежные с x (см. рис. 3.8). Поскольку — максимальный цикл, то

вершина v не смежна ни с a, ни с b (иначе можно было бы построить больший цикл). Рассмотрим теперь граф $G \setminus \{x\}$, который, очевидно, связен. В этом графе для каждой вершины $y \in C$, имеется $(v,\ y)$-путь. Выберем их этих путей кратчайший $(v,\ y^*)$-путь P^*. Учитывая, что C — максимальный цикл, имеем $y^* \neq a$, $y^* \neq b$. Кроме того, P^* не содержит вершин цикла C, отличных от y^*, так как иначе можно было бы построить более короткую цепь. Объединив цикл и путь P^*, получим искомый тэта-подграф. □

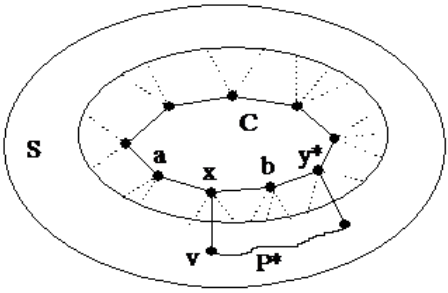


Рис. 3.8

Определение. *Степенной последовательностью графа* будем называть последовательность степеней его вершин.

Теорема 3.20. *Граф со степенной последовательностью* $d_1 \leq d_2 \leq \ldots \leq d_n$ *является гамильтоновым, если для всякого* k, *удовлетворяющего неравенствам* $1 \leq k < \dfrac{n}{2}$, *истинна импликация*

$$(d_k \leq k) \Rightarrow (d_{n-k} \geq n - k).$$

Доказательство. Доказательство этой теоремы здесь не приводится. □

В радиоэлектронике и микроэлектронике, в задачах проектирования железнодорожных и других путей, где нежелательны пресечения и переезды, возникает понятие плоского графа. *Плоским графом* называется граф, вершины которого являются точками плоскости, а ребра — непрерывными плоскими линиями без самопересечений, соединяющими соответствующие вершины так, что никакие два ребра не имеют общих точек, кроме инцидентной им обоим вершины. Любой граф, изоморфный плоскому графу, называется *планарным* графом.

Следующая теорема — один из наиболее сильных результатов, касающихся гамильтоновых графов.

Теорема 3.21. *Всякий 4-связный планарный граф является гамильтоновым.*

Доказательство. Весьма сложное доказательство этой теоремы здесь не приводится. □

Пусть $G =< V,\ E >$ — связный граф, u, v — две его несовпадающие вершины. Длина кратчайшего $(u,\ v)$-пути (он, естественно, является простым путем) называется *расстоянием между вершинами* u и v и обозначается через $d(u,\ v)$. Понятие расстояния между вершинами в связном графе позволяет определить k-ую степень графа. Пусть G — связный граф, k — натуральное число. Граф G^k имеет то же множество вершин, что и G, несовпадающие вершины u и v смежны в графе G^k тогда и только тогда, когда для графа G верно неравенство $d(u,\ v) \leq k$. Очевидно, что если $k \geq |V| - 1$, то G^k — полный граф.

Интуитивно ясно, что если граф порядка $n = |V| \geq 3$ связен, то при достаточно больших k граф G^k гамильтонов. Приведем без доказательства две теоремы, касающиеся гамильтоновости степеней графа.

Теорема 3.22. *Если граф G порядка $n = |V| \geq 3$ связен, то G^3 — гамильтонов граф.*

Теорема 3.23. *Если G — двусвязный граф порядка $n = |V| \geq 3$, то G^2 — гамильтонов граф.*

3.9 Нахождение кратчайших путей в графе

В этом разделе мы будем рассматривать ориентированные графы $G =< V,\ E >$, ребрам которых приписаны веса. Это означает, что каждому ребру $< u,\ v > \in E$ поставлено в соответствие вещественное число $c(u,\ v)$, называемое весом данного ребра. Полагаем, что $(u,\ v) = \infty$, если $< u,\ v > \notin E$.

Если $S =< v_0,\ v_1,\ \ldots,\ v_p >$ — путь в G, то его длина определяется как сумма

$$\sum_{i=1}^{p} c(v_{i-1}, v_i).$$

(Отметим, что если в произвольном графе мы примем вес каждого ребра равным единице, то получим обычное определение длины пути как числа ребер).

Нас будет интересовать нахождение кратчайшего пути между фиксированными вершинами $v,\ t \in V$. Длину такого кратчайшего пути $d(v,\ t)$ и будем называть расстоянием от v до t. Отметим, что если каждый цикл нашего графа имеет положительную длину, то кратчайший путь будет всегда простым.

Во многих приложениях достаточно находить кратчайшие пути между двумя конкретными вершинами, однако, неизвестен алгоритм, который решал бы эту задачу эффективнее (в худшем случае), чем лучший из известных алгоритмов для нахождения кратчайших расстояний от одной вершины v_0, называемой источником, до всех остальных вершин.

Задача нахождения кратчайших путей от одного источника решается для трех случаев.

1. Ориентированный граф без циклов с отрицательной длиной (сложность алгоритма — $O(n^3)$, где n — число вершин).

2. Веса всех ребер неотрицательны (сложность алгоритма — $O(n^2)$).

3. Граф без циклов (сложность алгоритма — $O(n^2)$).

3.9.1 Алгоритм нахождения расстояния от источника до всех остальных вершин в ориентированном графе с неотрицательными весами рёбер

Исходные данные: $G =< V,\ E >$ — ориентированный, связный, конечный граф с неотрицательными весами ребер.

v_0 — источник, $v_0 \in V$.

$C = ||c(u,\ v)||;\ u, ; v \in V;\ c(u,\ v) \geq 0$ — матрица весов ребер.

Результат: расстояния от источника до всех вершин графа $D[v] = d(v_0,\ v)$, $v \in V$.

Метод. Строим такое подмножество S множества вершин графа, $S \subseteq V$, что кратчайший путь из источника в каждую вершину $v \in S$ целиком лежит в S.

$S := \{v_0\}$;
$D[v] := c(v_0,\ v)$;
$D[v_0] := 0$;
while $V \neq S$ **do**

begin
выбрать вершину $u \in V \setminus S$, для которой $D(u) = \min\limits_{v \in V} D(v)$;
$S := S \cup \{u\}$;
для всех $v \in V \setminus S$ $D[v] := \min(D[v], D[v] + C[u,\ v])$
end

Чтобы показать корректность алгоритма, надо доказать индукцией по размеру множества S, что для каждой вершины $v \in S$ число $D[v]$ равно длине кратчайшего пути из v_0 в v. Более того, для всех $v \in V \setminus S$ число $D[v]$ равно длине кратчайшего пути из v_0 в v, лежащего целиком (если не считать саму вершину v) в S.

Базис индукции. Пусть $|S| = 1$. Кратчайший путь из v_0 в себя имеет длину 0, а путь из v_0 в v, лежащий целиком (исключая v) в S, состоит из единственного ребра $< v_0,\ v >$.

Шаг индукции. Пусть u — узел для которого $D(u) = \min\limits_{v \in V} D(v)$.

Если число $D[u]$ не равно длине кратчайшего пути из v_0 в u, то должен быть более короткий путь P. Этот путь должен содержать вершину, отличную от u и не принадлежащую S. Пусть v — первая такая вершина на пути P из вершины v_0 в вершину u (смотрите рис. 3.9).

Но тогда расстояние от v_0 до v меньше $D[u]$, а кратчайший путь в вершину v, целиком (исключая сам узел v) лежит в S. Следовательно, по предположению индукции, $D[v] < D[u]$ в момент выбора u; таким образом, мы пришли к противоречию. Отсюда заключаем, что такого пути P нет и $D[u]$ — длина кратчайшего пути из v_0 в u.

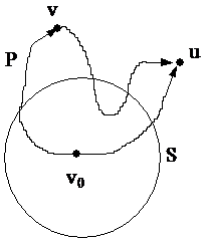


Рис. 3.9

Второе утверждение (о том, что $D[u]$ остается корректным) очевидно ввиду последнего оператора присваивания в алгоритме.

3.10 Максимальный поток в сети

Под сетью будем понимать пару $S =< G,\ c >$, где $G =< V,\ E >$ — произвольный ориентированный граф, а $c : E \to R$ — функция, которая каждому ребру $< u,\ v >$ ставит в соответствие неотрицательное вещественное число $(u,\ v)$, называемое пропускной способностью ребра.

Если для $f : E \to R$ $f(u,\ v)$ мы интерпретируем как поток из u в v, то величина $D_f(v)$,

$$D_f(v) = \sum_{u:<v,\ u>\in E} f(v,\ u) - \sum_{u:<u,\ v>\in E} f(u,\ v),$$

определяет «количество потока», выходящего из вершины v.

Если $D_f(v) > 0$, то вершина v называется источником, если $D_f(v) < 0$, то вершина v называется стоком. Для большинства вершин $D_f(v) > 0$.

Выделим в нашей сети две вершины — источник s и сток t. Под потоком из вершины s в вершину t в сети S будем понимать произвольную функцию $f : E \to R$, для которой выполняются условия:

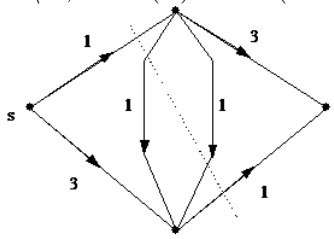
1. $0 \le f(u,\ v) \le c(u,\ v)$ для всех рёбер $< u,\ v >\in E$;
2. $D_f(v) = 0$ для всех $v \in V \setminus \{s,\ t\}$.

Величину $W(f) = D_f(s)$ будем называть величиной потока.

Такой поток может описывать, например, поведение газа или жидкости в трубопроводе, потоки автомобилей, движение по железной дороге, передачу информации в информационной сети.

Мы будем интересоваться нахождением максимального потока в сети.

Под *разрезом* () сети S, соответствующим подмножеству $A \subseteq V$ $(A \neq \varnothing,$ $A \neq V)$ мы понимаем множество ребер $< u,\ v >\in E$, таких, что $u \in A$, $v \in V \setminus A$, т.е. $P(A) = E \cap (A \times (V \setminus A))$.



Разрез — это линия или множество линий, которые полностью разделяют источник и сток.

Для произвольного потока f в сети S поток через разрез $P(A)$ определяется естественным образом:

$$f(A, V \setminus A) = \sum_{e=<u,\ v>\in P(A)} f(u,\ v).$$

Лемма 3.24. *Если $s \in A$ и $t \in V \setminus A$, то для произвольного потока f из s в t имеет место соотношение*

$$W(f) = f(A, V \setminus A) - f(V \setminus A, A)$$

В общем виде лемма говорит, что общее количество потока можно измерить в произвольном разрезе, отделяющем s от t.

В частности, если $A = V \setminus \{t\}$, получаем в этом случае из леммы:

$$D_f(s) = W(f) = f(V \setminus \{t\},\ \{t\}) - f(\{t\},\ V \setminus \{t\}) =$$
$$= -(f(\{t\},\ V \setminus \{t\}) - f(V \setminus \{t\},\ \{t\})) = -D_f(t),$$

что выражает интуитивно понятный факт: в сток входит в точности такое количество потока, какое выходит из источника.

Доказательство. Просуммируем уравнения $D_f(v) = 0$ для всех $v \in A$. Эта сумма состоит из некоторого количества слагаемых $f(u,\ v)$ со знаком $+$ или $-$, причем хотя бы одна из вершин принадлежит . Если обе вершины принадлежат , то $f(u,\ v)$ появляется со знаком плюс в $D_f(u)$ и со знаком минус в $D_f(v)$, что в сумме дает 0.

Каждое из слагаемых $f(u,\ v),\ u \in A$ появляется в точности один раз со знаком плюс в $D_f(u)$, что в сумме дает $f(A,\ V \setminus A)$. Аналогичные слагаемые $f(u,\ v),\ u \in V \setminus A,\ v \in A$, отвечают за слагаемое $f(V \setminus A,\ A)$ С другой стороны, сумма равна $D_f(s) = W(f)$, ибо $D_f(s) = 0$ для каждого $v \in A \setminus \{s\}$. □

Определим пропускную способность разреза $P(A)$ следующим способом:

$$C(A,\ V \setminus A) = \sum_{e=<u,\ v>\in P(A)} c(u,\ v).$$

Под минимальным разрезом, разделяющим s и t, будем понимать произвольный разрез $P(A)$, $s \in A$, $t \in V \setminus A$ с минимальной пропускной способностью. Фундаментальным фактом теории потоков в сетях является классическая теорема о максимальном потоке и минимальном разрезе.

Теорема 3.25. *Величина каждого потока из s в t не превосходит пропускной способности минимального разреза, разделяющего s и t, причем существует поток, достигающий этого значения.*

Доказательство. Пусть $P(A)$ — минимальный разрез. В силу леммы для произвольного потока f имеем

$$W(f) = f(A,\, V \setminus A) - f(V \setminus A,\, A) \le f(A,\, V \setminus A) =$$
$$= \sum_{e \in P(A)} f(e) \le \sum_{e \in P(A)} c(e) = C(A,\, V \setminus A).$$

Существование потока, для которого указанное неравенство переходит в равенство (такой поток, очевидно, максимален), гораздо более глубокий факт, который здесь мы доказывать не будем. □

Литература

[1] Стенли Р. Перечислительная комбинаторика. М.: Мир, 1990.

[2] Липский В. Комбинаторика для программистов. М.: Мир, 1988.

[3] Рыбников К.А. Введение в комбинаторный анализ. М.: МГУ,1985.

[4] Гаврилов Г.И., Сапоженко А.А. Задачи и упражнения по курсу дискретной математики. М.: Наука, 1992.

[5] Риордан Дж. Введение в комбинаторный анализ. М.: ИЛ, 1963.

[6] Холл М. Комбинаторика. М.: Мир, 1970.

[7] Мендельсон Э. Введение в математическую логику.- М.: Наука, 1976.

[8] Дискретная математика и математические вопросы кибернетики. Под ред. С.В.Яблонского, О.В.Лупанова, Т.1, М.; Наука, 1974.

[9] Яблонский С.В. Введение в дискретную математику. М.: Наука, 1979.

[10] Оре О. Теория графов. М.: Наука, 1968.

[11] Кристофидис Н. Теория графов. Алгоритмический подход. М.: Мир, 1987.

[12] Емеличев В.А., Мельников О.И. и др. Лекции по теории графов. М.: Наука, 1990.

[13] Уилсон Р.Дж. Введение в теорию графов. М.: Мир, 1977.

[14] Харари Ф. Теория графов. М.: Мир,1973.