Библиотека
«Математическое просвещение»
Выпуск 38

С. Б. Гашков

СЛОЖЕНИЕ ОДНОБИТНЫХ ЧИСЕЛ

Треугольник Паскаля, салфетка Серпинского и теорема Куммера

Издательство Московского центра
непрерывного математического образования
Москва • 2014

УДК 511.2
ББК 22.131
Г24

Гашков С. Б.

Г24 Сложение однобитных чисел. Треугольник Паскаля, салфетка Серпинского и теорема Куммера. — М.: МЦНМО, 2014. — 40 с.

ISBN 978-5-4439-0145-9

В книге рассказывается о любопытной связи задачи о сложении чисел в двоичной записи с алгеброй логики, многочленами Жегалкина, треугольником Паскаля, салфеткой Серпинского и теоремой Куммера о делимости биномиальных коэффициентов. Все необходимое для понимания разъясняется. Брошюра является расширенным вариантом лекции, прочитанной на Малом мехмате в МГУ им. Ломоносова 6 апреля 2013 г.

ББК 22.131

ISBN 978-5-4439-0145-9

Памяти моих родителей

1. Так ли просто сложение?

Всем известно, что самая простая часть школьной математики — это арифметика. А в ней самая простая операция — сложение. Но слово *сложность* родственно, очевидно, слову *сложение*. Странно, не правда ли? Что может быть сложного в сложении?

Мы постараемся далее показать, что сложение, даже в самой простой позиционной системе — двоичной — не так уж и просто, как это кажется на первый взгляд. Совсем непростой оказывается даже, казалось бы, простейшая задача — сложение одноразрядных (или, как говорят в программировании, однобитных) чисел[1].

Сформулируем эту задачу в общем виде.

Задача. Нужно сложить n чисел $x_i = 0$ или 1 и результат получить в двоичной записи, т. е. найти такие двоичные цифры y_i, чтобы выполнялось равенство

$$x_1 + \ldots + x_n = (y_m \ldots y_0)_2 = y_m 2^m + \ldots + 2y_1 + y_0, \quad 2^m > n \geqslant 2^{m-1}.$$

Начнем разбираться с простейших частных случаев. Тем, кто незнаком с двоичной системой, рекомендуем заглянуть в книжки [1–3]. Впрочем, все, что нужно, далее будет объяснено.

Рассмотрим вначале случай $n = 2$. Так как

$$0 + 0 = 0 = (00)_2, \quad 0 + 1 = 1 = (01)_2, \quad 1 + 1 = 2 = (10)_2,$$

результаты сложения можно задать таблицей 1.

Таблица 1. Сложение двух однобитных чисел

x_1	x_2	y_1	y_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Случай $n = 3$ чуть более сложен. И таблица сложения имеет вдвое больший размер (см. таблицу 2).

[1] Бит — это английское, давно уже ставшее международным, сокращение слов *binary digit — двоичная цифра*.

Таблица 2. Сложение трех однобитных чисел

x_1	x_2	x_3	y_1	y_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Ясно, что если, например, $n = 15$, то в качестве m можно взять 3 и соответствующая таблица будет иметь $2^{15} = 32768$ строк. А в строке будет $n + m + 1 = 19$ битов, значит, в компьютере эта таблица будет занимать не менее 76 килобайт памяти (а может, и больше, — все зависит от того, как ее в памяти хранить). Компьютер с ней справится, но вручную с такой таблицей работать невозможно. А если $n = 30$, не поможет и компьютер. Очевидно, язык таблиц для нашей задачи (и подобных ей) неудобен. Удобным языком является язык формул алгебры логики. Хотите получить представление о том, что это такое, — читайте следующий раздел.

2. Двоичная система в математике и электронике

Главное достоинство двоичной системы — простота алгоритмов арифметических операций. Таблица умножения в ней совсем не требует ничего запоминать: ведь любое число, умноженное на нуль, равно нулю, а умноженное на единицу — равно самому себе.

Таблица сложения в двоичной системе чуть сложнее таблицы умножения (в отличие от десятичной системы), потому что $1+1=10$ и возникает перенос в следующий разряд. В общем виде операцию сложения однобитовых чисел можно записать в следующем виде: $x + y = 2w + v$, где w, v — биты результата.

Внимательно посмотрев на таблицу 3, можно заметить, что бит переноса w — это просто произведение xy, потому что он равен единице, лишь когда x и y равны единице. Произведение (конъюнкция) обычно обозначается символом &. А вот бит v равен $x + y$, за исключением случая $x = y = 1$, когда он равен не 2, а 0. Операцию,

Таблица 3. Сложение в двоичной системе

x	y	w	v
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

с помощью которой по битам x, y вычисляют бит v, называют по-разному. Мы будем использовать для нее название *сложение по модулю* 2 и символ $\oplus$. Таким образом, сложение битов выполняется фактически не одной, а двумя операциями — конъюнкцией и сложением по модулю 2.

Если отвлечься от технических деталей, то именно с помощью этих операций и выполняются все операции в компьютере.

Кроме этих операций, часто используется операция *дизъюнкция*, обозначаемая далее $x \vee y$. Она отличается от операции $x \oplus y$ только тем, что $1 \vee 1 = 1$, а $1 \oplus 1 = 0$. Также полезна операция *отрицания* $\neg x = 1 - x = 1 \oplus x$. Указанные операции связаны между собой множеством тождеств.

Задача 1. Докажите тождества:

$$x \vee y = \neg(\neg x \,\&\, \neg y),$$
$$x \,\&\, y = \neg(\neg x \vee \neg y),$$
$$x \oplus y = (x \,\&\, \neg y) \vee (y \,\&\, \neg x),$$
$$x \vee y = x \oplus y \oplus (x \,\&\, y).$$

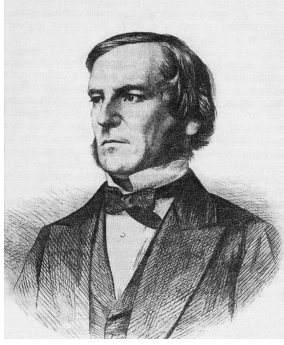
Отрицание обозначается также чертой сверху, например, вместо $\neg(x \,\&\, y \,\&\, z)$ пишут $\overline{x \,\&\, y \,\&\, z}$. Можно выразить указанные операции, называемые логическими, через обычные арифметические операции. Конъюнкция, например, просто совпадает с обычным умножением, поэтому ее при записи часто обозначают точкой или вообще пропускают.

Задача 2. Докажите тождества:

$$x \oplus y = x + y - 2xy,$$
$$x \vee y = x + y - xy,$$
$$x \vee y = \max(x, y) = 1 - \min(1 - x, 1 - y),$$
$$x \,\&\, y = \min(x, y) = 1 - \max(1 - x, 1 - y).$$

Двоичная система и логические операции

Почему эти операции называют логическими? Потому что если сопоставить каждому высказыванию A его «истинностное» значение $|A| = 0$, если A ложно, и $|A| = 1$, если A истинно, то истинностное значение составного высказывания «A и B» можно выразить[2] через истинностные значения высказываний A, B по формуле



Джордж Буль

$$|A \text{ и } B| = |A| \,\&\, |B|$$

и, аналогично,

$$|A \text{ или } B| = |A| \vee |B|.$$

В последней формуле мы предполагали, что высказывание «A или B» будет истинным и тогда, когда оба высказывания A и B истинны. Иногда союз «или» понимают в несколько другом, разделительном, смысле: составное высказывание «A или B» считается истинным только в случае, если ровно одно из высказываний A, B истинно, но не оба сразу. В этом случае для вычисления истинностного значения «A или B» можно использовать формулу

$$|A \text{ или } B| = |A| \oplus |B|.$$

3. Подсчет числа единиц в двоичной строке

Итак, нам нужно сложить n чисел $x_i = 0, 1$ и результат получить в двоичном виде, т. е.

$$x_1 + \ldots + x_n = (y_m \ldots y_0)_2, \quad 2^{m+1} > n \geqslant 2^m.$$

Как выразить y_i через $x_1, \ldots, x_n$? Очевидно, что y_i выражается через $x_1, \ldots, x_n$ однозначно, т. е. является некоторой функцией $f_i(x_1, \ldots, x_n)$ от переменных x_j. Эти переменные принимают значения нуль и единица, и сама функция тоже принимает только такие значения. Такие

[2] Впервые это сделал Джордж Буль (1815–1864) — английский, а точнее ирландский, математик, один из основателей математической логики. Был профессором университета в городе Корк и отцом шести дочерей, одна из которых стала математиком, другая — первой женщиной-профессором химии в Англии, а еще одна — писательницей, известной под именем Этель Лилиан Войнич.

Таблица 4. Булевы функции, связанные со сложением двух однобитных чисел

x_1	x_2	f_1	f_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

функции называют функциями алгебры логики или булевыми функциями (в честь Дж. Буля)[3].

Рассмотрим вначале простейший случай $n = 2$. Так как

$$0 + 0 = 0 = (00)_2, \quad 0 + 1 = 1 = (01)_2, \quad 1 + 1 = 2 = (10)_2,$$

функции f_1, f_0 от переменных x_1, x_2 можно задать таблицей, по существу совпадающей с таблицей 4.

Но более компактно вместо таблицы определить функции формулами $f_1(x_1, x_2) = x_1 \,\&\, x_2$, $f_0(x_1, x_2) = x_1 \oplus x_2$. Случай $n = 3$ чуть более сложен. В нем должно выполняться тождество

$$x_1 + x_2 + x_3 = 2y_1 + y_0, \quad y_1 = f_1(x_1, x_2, x_3), \quad y_0 = f_0(x_1, x_2, x_3)$$

для некоторых булевых функций f_0, f_1. Эти функции можно задать таблицей 5. Для того чтобы ускорить ее заполнение, можно заметить, что функции f_i симметрические, т. е. они не меняют значений при любой перестановке переменных (ведь сумма $x_1 + x_2 + x_3$ не меняется при перестановке переменных.)

Таблица 5. Булевы функции для сложения трех однобитных чисел

x_1	x_2	x_3	f_1	f_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

[3] В определенном смысле булевы функции являются самыми простыми функциями из всех, встречающихся в математике. Они существенно проще, чем, например, тригонометрические функции.

Глядя на таблицу, можно заметить, что функция f_0 обладает таким свойством: она равна единице, если в наборе (x_1, x_2, x_3) нечетное число единиц (одна или три), в противном случае она равна нулю. Другими словами, она получается из обычной суммы $x_1 + x_2 + x_3$ с помощью замены результата суммирования на остаток от его деления на два. Такую манипуляцию называют приведением по модулю два и ее результат обозначают как $x_1 + x_2 + x_3 \mod 2$.

Задача 3. Проверьте, что

$$x_1 + x_2 + x_3 \mod 2 = (x_1 \oplus x_2) \oplus x_3 = x_1 \oplus (x_2 \oplus x_3)$$

и в этих тождествах переменные можно произвольным образом переставлять. Докажите, что аналогичные тождества справедливы для любого числа n булевых переменных x_i (т. е. переменных, принимающих только значения 0 или 1).

Поэтому функции $x_1 + x_2 + \ldots + x_n \mod 2$ обозначают для краткости $x_1 \oplus \ldots \oplus x_n$ (не уточняя порядок расстановки скобок, так как он несущественен). Значит, $f_0(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$.

Из таблицы также видно, что функция f_1 обладает следующим свойством: она равна единице, если в наборе (x_1, x_2, x_3) число единиц не меньше двух, иначе она равна нулю.

Задача 4. Проверьте, что тем же свойством обладает булева функция, определяемая формулой $x_1x_2 \vee x_1x_3 \vee x_2x_3$, и тем самым

$$f_1(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3.$$

В силу симметричности этих формул достаточно проверить это равенство на наборах (000), (100), (110), (111).

Задача 5. Проверьте, что функцию f_1 можно выразить (или, как еще говорят, реализовать) и формулой $f_1 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$.

Теперь можно догадаться, каким будет ответ и в общем случае. Так как формулы довольно громоздки, мы их приведем только для случая $n = 15$. Оказывается,

$$y_0 = x_1 \oplus \ldots \oplus x_{15},$$

$$y_1 = \sum_{1 \leqslant i < j \leqslant 15} x_i x_j = x_1x_2 \oplus x_1x_3 \oplus \ldots \oplus x_1x_{15} \oplus x_2x_{15} \oplus \ldots \oplus x_{14}x_{15},$$

$$y_2 = \sum_{1 \leqslant i < j < k < l \leqslant 15} x_i x_j x_k x_l = x_1x_2x_3x_4 \oplus x_1x_2x_3x_5 \oplus \ldots \oplus x_{12}x_{13}x_{14}x_{15},$$

$$y_3 = \sum_{1 \leqslant i_1 < \ldots < i_8 \leqslant 15} x_{i_1} \ldots x_{i_8} = x_1 \ldots x_8 \oplus x_1 \ldots x_7 x_9 \oplus \ldots \oplus x_8 \ldots x_{15}.$$

Приведенные формулы представляют из себя многочлены по модулю два степеней один, два, четыре, восемь. Они являются суммами по модулю два одночленов вида $x_{i_1} \ldots x_{i_m}$, не содержащих степеней переменных выше первой. Одночлены с высокими степенями переменных для реализации булевых функций не нужны, так как для булевых переменных справедливо тождество $x^2 = x$ и, как следствие, справедливы тождества $x^m = x$ при любом $m > 1$. Многочлены такого вида называются также многочленами Жегалкина[4]. Из приведенных формул видно, что использованные в них многочлены Жегалкина являются симметрическими (и по существу совпадают с многочленами, появляющимися в общей теореме Виета).

Первую формулу доказать просто. Действительно, $y_0 = 1$ тогда и только тогда, когда в наборе $(x_1 \ldots x_{15})$ число единиц нечетно. Но тогда и сумма

$$x_1 + \ldots + x_{15} = 8y_3 + 4y_2 + 2y_1 + y_0$$

нечетна, а это возможно только при $y_0 = 1$. Остальные формулы доказать сложнее. Это можно сделать разными способами. Мы укажем один из них, который приведет нас к интересным задачам из комбинаторики.

4. Треугольник Паскаля и салфетка Серпинского

Треугольник Паскаля состоит из чисел $\binom{n}{k}$, называемых биномиальными коэффициентами. В его n-й строке стоят коэффициенты $\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}$, которые получаются после раскрытия скобок и приведения подобных членов в формуле $(1+x)^n$, а именно

$$(1+x)^n = \binom{n}{0} + \binom{n}{1} x + \ldots + \binom{n}{n} x^n.$$

Открытие этого треугольника приписывается знаменитому французскому математику, физику и религиозному философу Блезу Паскалю (хотя в том или ином виде подобные таблицы были известны

[4] В честь профессора мехмата МГУ И. И. Жегалкина (1869–1947), доказавшего в 1927 г., что любую булеву функцию можно единственным способом реализовать такими многочленами.

Блез Паскаль

и до него, например, китайцам, да и в Западной Европе до Паскаля подобную таблицу, только прямоугольную, приводил в своей книге итальянец Никколо Тарталья — тот самый, который один из первых в мире научился решать кубические уравнения). Разных задач про биномиальные коэффициенты существует великое множество, и далее мы коснемся только тех, которые нам непосредственно понадобятся. Если вы хотите еще что-нибудь прочитать про биномиальные коэффициенты, то можете взять, например, книги [4, 8].

Очевидно, что $\binom{n}{0}=1$, $\binom{n}{n}=1$. Раскрывая скобки и приравнивая коэффициенты в обеих частях тождества

$$\binom{n+1}{0}+\binom{n+1}{1}x+\ldots+\binom{n+1}{n+1}x^n=(1+x)^{n+1}=(1+x)^n(1+x)=$$
$$=\left(\binom{n}{0}+\binom{n}{1}x+\ldots+\binom{n}{n}x^n\right)(1+x),$$

получаем тождество Паскаля

$$\binom{n+1}{k+1}=\binom{n}{k+1}+\binom{n}{k},$$

которое и лежит в основе треугольника Паскаля. С его помощью можно вычислять биномиальные коэффициенты, используя только операцию сложения (см. таблицу 6).

Таблица 6. Первые 11 строк треугольника Паскаля

1										
1	1									
1	2	1								
1	3	3	1							
1	4	6	4	1						
1	5	10	10	5	1					
1	6	15	20	15	6	1				
1	7	21	35	35	21	7	1			
1	8	28	56	70	56	28	8	1		
1	9	36	84	126	126	84	36	9	1	
1	10	45	120	210	252	210	120	45	10	1

Задача 6. Последовательно используя тождество Паскаля, проверьте, что всегда выполняется равенство

$$\binom{n}{k} = \frac{n!}{k!(n-k)!},$$

где $n! = n \cdot (n-1) \ldots 2 \cdot 1$.

Задача 7. Проверьте, что $\binom{n}{k} = \binom{n}{n-k} = \frac{n(n-1)\ldots(n-k+1)}{k!}$.

Заметим, что из n скобок $(1+x)\cdots(1+x)$ выбрать k раз символ x и $n-k$ раз единицу можно в точности $\binom{n}{k}$ способами, так как

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x + \ldots + \binom{n}{n}x^n.$$

Поэтому выбрать из любых n различных символов $x_1, \ldots, x_n$ в точности k различных символов $x_{i_1}, \ldots, x_{i_k}$ тоже можно ровно $\binom{n}{k}$ способами.

Задача 8. Сколько слагаемых в многочленах Жегалкина, выражающих y_0, y_1, y_2, y_3?

Ответ: у многочлена y_i в точности $\binom{15}{2^i}$ слагаемых.

Обозначим через

$$\|x\| = x_0 + \ldots + x_{15}$$

число единиц в наборе $(x_0, \ldots, x_{15})$.

Задача 9. Докажите, что значение $y_i = f_i(x_1, \ldots, x_{15})$ равно

$$\binom{\|x\|}{2^i} \bmod 2$$

(здесь и далее удобно считать, что $\binom{n}{k} = 0$ при $k > n$).

Указание. Число единиц в сумме для y_i равно $\binom{\|x\|}{2^i}$.

Далее нас будут интересовать не сами биномиальные коэффициенты, а их значения по модулю два. Удобный способ их вычислить состоит в том, что вместо вычисления биномиальных коэффициентов и деления их на два с остатком можно построить треугольник Паскаля по модулю два так, как показано в таблице 7.

Глядя на таблицу[5] 7, легко заметить, что первый ее столбец состоит только из единиц, во втором столбце нули и единицы чередуются, в третьем столбце нули и единицы стоят парами, и эти пары

[5] Она квадратная, а не треугольная, но ее половина, лежащая выше диагонали, заполнена нулями, что соответствует принятому выше соглашению $\binom{n}{k}=0$ при $n<k$, так что фактически это тот же треугольник Паскаля, только вычисленный по модулю два.

Таблица 7. Треугольник Паскаля по модулю два

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

чередуются (т. е. четверка 0011 повторяется периодически), в пятом столбце нули и единицы группируются по четыре (т. е. повторяется периодически восьмерка битов 00001111), в девятом столбце нули и единицы группируются по восемь (т. е. повторяется периодически шестнадцать битов 0000000011111111) и так далее. Уже этого наблюдения достаточно, чтобы обосновать то, что указанные выше многочлены вычисляют в точности двоичные цифры интересующей нас суммы 15 битов[6]. Для строгого доказательства полезно обратить также внимание на тот факт, что треугольник Паскаля по модулю два размера 2^n (обозначим его T_n) состоит из трех равных треугольников T_{n-1} (треугольников Паскаля по модулю два вдвое меньшего размера) и одного треугольника, заполненного нулями, как показано на рис. 1.

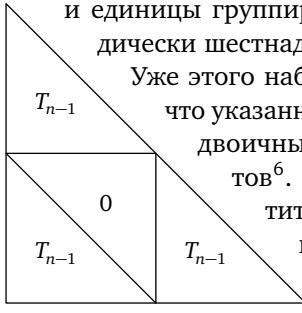


Рис. 1. Фрактальная структура треугольника Паскаля по модулю два

Для желающих строго обосновать сделанные наблюдения предлагаем решить следующие задачи.

Задача 10. Докажите, что все числа в строке с номером $2^n - 1$ треугольника Паскаля нечетны (равны единице по модулю два).

[6] Древние индусы вместо доказательств писали просто — «Смотри!».

Задача 11. Докажите, что все числа в 2^n-й строке треугольника Паскаля, кроме первого и последнего, четны.

Из обоснованной выше рекурсивной конструкции можно вывести следующее утверждение.

Задача 12. Пусть $n = (n_m \dots n_0)_2$ — двоичная запись числа n. Докажите, что

$$\binom{n}{2^k} \bmod 2 = n_k, \quad k = 0, \dots, m.$$

Это утверждение и позволяет легко доказать, что указанные многочлены Жегалкина реализуют функции f_i. Действительно, достаточно заметить, что $\|x\| = 8y_3 + 4y_2 + 2y_1 + y_0$, а значение

$$f_i(x_1, \dots, x_{15}) = \binom{\|x\|}{2^i} \bmod 2 = \binom{8y_3 + 4y_2 + 2y_1 + y_0}{2^i} \bmod 2 = y_i,$$
$$i = 0, \dots, 3.$$

Аналогично проводится доказательство и в общем случае.

Следующая задача обобщает предыдущую.

Задача 13. Пусть $n = (n_m \dots n_0)_2$ — двоичная запись числа n, а $k = (k_m \dots k_0)_2$ — двоичная запись числа k. Докажите, что $\binom{n}{k}$ нечетно тогда и только тогда, когда $k_i \leqslant n_i$ при всех $i = 0, \dots, m$.

Из предыдущей задачи легко следует

Задача 14. Докажите, что количество нечетных чисел в любой строке треугольника Паскаля равно степени двойки.

Следующая задача, как и предыдущая, нам для обоснования формул для сложения уже не нужна, но она любопытна и сама по себе.

Задача 15. Докажите, что число единиц в треугольнике Паскаля по модулю два с основанием длины 2^n равно 3^n.

Указание. Треугольник T_n состоит из трех треугольников T_{n-1} и треугольника, заполненного нулями.

Можно еще заметить, глядя на таблицу 7, что в девятом столбце верхняя половина состоит сплошь из нулей, а нижняя — сплошь из единиц. Это значит, что булева функция $f_3(x_1, \dots, x_{15}) = 1$ тогда и только тогда, когда $\|x\| = x_1 + \dots + x_{15} \geqslant 8$, значит, в формуле для нее можно заменить везде операцию $\oplus$ сложения по модулю два на операцию $\vee$ дизъюнкции и вместо многочлена Жегалкина

представить f_3 в виде

$$f_3(x_1,\ldots,x_{15}) = \bigvee_{1\leqslant i_1<\ldots<i_8\leqslant 15} x_{i_1}\ldots x_{i_8} = \\ = x_1\,\&\ldots\&\,x_8 \vee x_1\,\&\ldots\&\,x_7\,\&\,x_9 \vee \ldots \vee x_8\,\&\ldots\&\,x_{15}.$$

Приведенная выше формула является так называемой монотонной дизъюнктивной нормальной формой, а сама функция $f_3(x_1,\ldots,x_{15})$ является монотонной булевой функцией. По определению функция монотонна, если при увеличении любой ее переменной значение функции не уменьшается.

Задача 16. Проверьте, что остальные функции f_0, f_1, f_2 от переменных $x_1,\ldots,x_{15}$ не монотонны.

Разумеется, приведенные выше утверждения верны и в общем случае (см. задачу 23 на с. 20).

Треугольник Паскаля по модулю два тесно связан с так называемым треугольником Серпинского[7]. Треугольник Серпинского определяется следующим образом. Правильный треугольник разрезается средними линиями на четыре равных треугольника и средний из них удаляется. С каждым из оставшихся трех треугольников поступаем аналогично и этот процесс продолжаем до бесконечности. Те точки, которые не будут удалены из треугольника, образуют треугольник Серпинского (называемый еще иногда салфеткой Серпинского). Построение треугольника Серпинского осуществляется рекурсивным образом и проиллюстрировано на рис. 2.

Вацлав Серпинский

Окончательно салфетка Серпинского выглядит приблизительно как на рис. 3. Если на выполнение $(n+1)$-го шага рекурсии, т. е. на вырезание треугольных дырок в каждом из получившихся на предыдущем шаге 3^n треугольников, тратить $(1/2^n)$-ю долю секунды, то за две секунды все будет закончено!

[7] Вацлав Франциск Серпинский (1882–1969) — выдающийся польский математик и популяризатор науки. Учился в Варшавском университете, где тогда преподавание велось на русском языке. Дипломную работу по теории чисел защитил под руководством Г. Ф. Вороного.

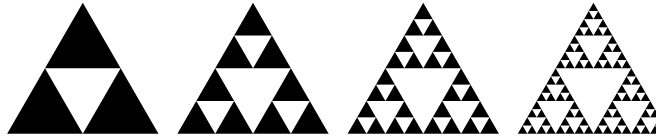


Рис. 2. Построение треугольника Серпинского.
Первые четыре шага рекурсии

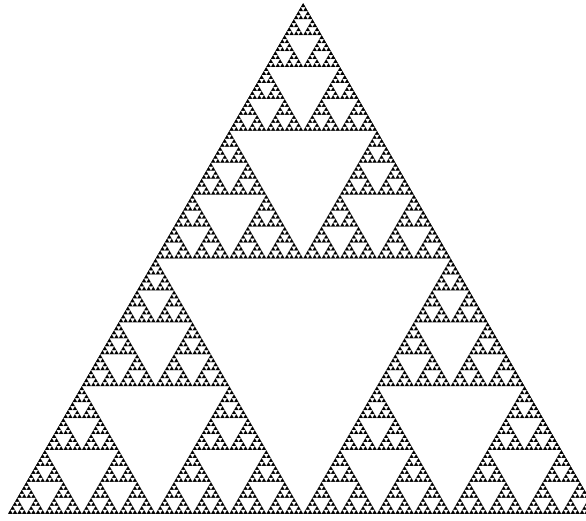


Рис. 3. Треугольник Серпинского

Серпинский придумал ее еще в 1915 г. вместе с другими подобными множествами, например, квадратом (или ковром) Серпинского, кубом Серпинского, тетраэдром Серпинского — см. об этом хорошую научно-популярную книгу [5]. Но наибольшую известность в широких кругах это множество получило на рубеже XX–XXI веков в связи с вошедшим в моду понятием фрактала. Оказалось, что салфетка и другие множества, построенные Серпинским, являются примерами так называемых фрактальных множеств. Они обладают различными любопытным свойствами. Например, площадь салфетки (точнее, мера Лебега[8] этого множества) равна нулю. Этот факт вытекает из следующего утверждения.

[8] Анри Лебег (1875–1941) — знаменитый французский математик, создатель теории меры и интеграла Лебега.

Задача 17. Докажите, что салфетка Серпинского накрывается 3^n равными треугольниками, площадь каждого из которых равна $1/4^n$ площади треугольника, из которого вырезали эту салфетку.

Салфетка Серпинского имеет дробную размерность по Хаусдорфу[9] (она равна $\log_2 3$, если кому интересно). Есть у этой салфетки и другие интересные особенности. Не будем углубляться в эти вопросы (заинтересованного читателя отошлем к многочисленным книгам про фракталы, например к книгам основателя и популяризатора этого направления Бенуа Мандельброта), а опять вернемся к элементарной комбинаторике.

5. Переносы при сложении двоичных чисел и теорема Куммера

В этом разделе мы изучим более глубоко вопрос, на какие степени двойки могут делиться биномиальные коэффициенты, и даже рассмотрим аналогичный вопрос о делимости их на степени заданного простого числа p, хотя непосредственно с задачей о сложении однобитных чисел это никак не связано.

Будем рассматривать позиционные числовые системы с произвольным основанием $b \geqslant 2$. Обозначим через $\nu_b(n)$ сумму всех b-ичных цифр в b-ичной записи числа n. Нам понадобится

Лемма 1. *Если b-ичная запись числа n оканчивается ровно m нулями, то*

$$\nu_b(n-1) + 1 - \nu_b(n) = (b-1)m,$$

и при прибавлении к числу $n-1$ единицы число m будет равно количеству произведенных во время этой операции переносов в следующий разряд (*возможно*, $m = 0$).

Доказательство. Заметим, что b-ичная запись числа $n-1$ оканчивается ровно m цифрами $b-1$. После прибавления единицы происходит m-кратный перенос в старшие разряды и получается число n, у которого все цифры, кроме $m+1$ последних, совпадают с такими же цифрами числа $n-1$, причем $(m+1)$-я от конца цифра на 1 больше такой же цифры числа $n-1$, а последние m цифр — нули. □

[9] Феликс Хаусдорф (1868–1942) — выдающийся немецкий математик. Его книга «Теория множеств» недавно была переиздана.

Рассмотрим еще одну лемму.

Эрнст Куммер

Лемма 2 (Куммер). *Количество переносов в следующий разряд при сложении чисел k и $n-k$ в b-ичной системе счисления равно*

$$s = \frac{\nu_b(n-k) + \nu_b(k) - \nu_b(n)}{b-1}.$$

Доказательство. Утверждение вытекает из леммы 1. Действительно, если переносов не происходило, то $s = 0$. Каждый перенос уменьшает один разряд в числе n на b и увеличивает следующий разряд на 1, в результате рассматриваемая величина возрастает на $b-1$, а значит, s возрастает на 1. □

Эта лемма была нужна Куммеру[10] для доказательства следующей теоремы.

Теорема 1 (Куммер). *Пусть* $\operatorname{ord}_p \binom{n}{k}$ *— показатель степени, в которой простое число p входит в разложение биномиального коэффициента*

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Тогда

$$\operatorname{ord}_p \binom{n}{k} = \frac{\nu_p(n-k) + \nu_p(k) - \nu_p(n)}{p-1},$$

где в правой части стоит количество переносов в следующий разряд при сложении чисел k и $n-k$ в p-ичной системе счисления.

Для доказательства этой теоремы полезна следующая формула Лежандра[11].

Теорема 2 (Лежандр). *Показатель степени, в которой простое число p входит в разложение факториала $n!$, равен*

$$\operatorname{ord}_p n! = \frac{n - \nu_p(n)}{p-1}.$$

[10] Эрнст Эдуард Куммер (1810–1893) — знаменитый немецкий математик, доказавший теорему Ферма во многих частных случаях и заложивший основы теории алгебраических чисел.

[11] Адриен Мари Лежандр (1752–1833) — выдающийся французский математик. Единственное дошедшее до нас изображение Лежандра — карикатура. То, что в течение долгого времени считали портретом А. М. Лежандра, является на самом деле портретом его однофамильца.

Доказательство. Действительно, если согласно предположению индукции

$$\operatorname{ord}_p(n-1)! = \frac{n-1-\nu_p(n-1)}{p-1} \quad \text{и} \quad \operatorname{ord}_p(n) = m,$$

то p-ичная запись числа n оканчивается ровно m нулями, значит, согласно лемме 1

$$\nu_p(n-1) + 1 - \nu_p(n) = (p-1)m,$$

откуда имеем

$$\operatorname{ord}_p n! = \operatorname{ord}_p(n-1)! + m = m + \frac{n-1-\nu_p(n-1)}{p-1} = \frac{n-\nu_p(n)}{p-1}.$$

База индукции $n < p$ очевидна, так как тогда $n = \nu_p(n)$. □

Задача 18. Докажите теорему Куммера.

Указание. Утверждение следует из формулы Лежандра, формулы для биномиального коэффициента и леммы 2.

С помощью теоремы Куммера легко решить задачи раздела 4, причем в более общем виде.

Задача 19. Пусть n записывается в p-ичной системе счисления при простом p в виде

$$n = (n_m \dots n_0)_p, \quad 0 \leqslant n_i < p, \quad i = 0, \dots, m = \lambda_p(n),$$

где $\lambda_p(n)$ — число, на единицу меньшее длины p-ичной записи числа n. Докажите, что тогда число биномиальных коэффициентов $\binom{n}{k}$, $0 \leqslant k \leqslant n$, не кратных p, равно $(n_0+1)\dots(n_m+1)$.

Указание. Согласно теореме Куммера число $\binom{n}{k}$ не кратно p тогда и только тогда, когда при сложении k и $n-k$ в p-ичной системе не происходит переносов в следующий разряд, т. е. когда

$$k = (k_m \dots k_0)_p, \quad 0 \leqslant k_i \leqslant n_i < p, \quad i = 0, \dots, m = \lambda_p(n).$$

Задача 20. Докажите, что n-я строка треугольника Паскаля состоит только из некратных простому p чисел тогда и только тогда, когда $n = p^m - 1$.

Еще одно решение этой задачи можно найти в [6].

Задача 21 (Московская олимпиада, 2012). Найдите число не кратных трем чисел в 2012-й строке треугольника Паскаля.

Указание. Достаточно записать число 2012 в троичной системе. Для ускорения вычислений удобно разложить его сначала в девяте-

ричной системе, а потом каждую цифру от 0 до 8 записать в троичной системе. Проще всего начать с младших разрядов. Самый младший равен остатку от деления 2012 на 9. Он, очевидно, равен 5. Отнимаем 5 и делим разность на 9. Получаем $2007 : 9 = 223$. С этим числом поступаем аналогично. Его младший разряд будет 7. После вычитания и деления на 9 получаем 24. В итоге имеем $2012 = 5 + 9(7 + 9(6 + 9 \cdot 2))$, значит, троичные цифры есть 2, 1, 1, 2, 0, 2, 2, поэтому число не кратных трем биномиальных коэффициентов равно $3 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 3 = 324$.

Задача 22 (Московская олимпиада, 2012). Докажите, что в строке треугольника Паскаля с номером 2012^{2011} количество не кратных 2011 чисел делится нацело на 2012.

Указание. Сначала надо проверить, что 2011 — простое число. Для этого достаточно установить, что оно не делится на

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43.$$

Потом надо разложить число $(p+1)^p$ в p-ичной системе при $p = 2011$. Полное разложение найти сложно, но нам достаточно найти несколько его цифр. Если n_i — эти цифры, то количество N не кратных p биномиальных коэффициентов $\binom{n}{k}$, где $n = (p+1)^p$, будет делиться на произведение чисел $(n_i + 1)$. Так как $2012 = 4 \cdot 503$, достаточно проверить, что среди цифр n_i есть хотя бы две единицы и число вида $503k - 1$, $k = 1, 2, 3$. Так как согласно формуле бинома

$$(1+p)^p = \sum_{k=0}^{p} p^k \binom{p}{k} =$$

$$= 1 + p^2 + p^3 \frac{p-1}{2} + p^4 \frac{(p-1)(p-2)}{6} + \ldots + p^{p-1} \frac{p-1}{2} + p^p + p^p$$

и все слагаемые, кроме первых трех, кратны p^4, это число при делении на p^4 равно $1 + p^2 + p^3 \frac{p-1}{2}$, поэтому младшие разряды его p-ичной записи равны $1, 0, 1, (p-1)/2 = 1005$, и этого уже достаточно, чтобы сделать вывод о делимости числа N на $(1+1)(1+1)(1005+1) = 4 \cdot 1006 = 2 \cdot 2012$. Так как сумма всех слагаемых, кроме последних двух, меньше p^p, старший разряд равен 2, поэтому N делится даже на $6 \cdot 2012$. Другое решение, в котором некоторые детали опущены, имеется в [7]. В заключение сообщим, что много задач о делимости биномиальных коэффициентов, в том числе и теоремы Куммера и Лежандра, можно найти в [8].

Задача 23. При каких n функция

$$f(x_1,\dots,x_n)=\sum_{1\leqslant i_1<\dots<i_m\leqslant n} x_{i_1}\dots x_{i_m},\quad m=\left\lceil\frac{n}{2}\right\rceil,$$

будет монотонной? Знак $\lceil x\rceil$ означает наименьшее целое число, не меньшее, чем x.

Ответ: при $n=2^k-1$, $k=1,2,3,\dots$

Указание. Если $2^{k-1}\leqslant l\leqslant 2^k-1$, то при сложении в двоичной системе чисел $m=2^{k-1}$ и $l-m<2^{k-1}$ переносов не происходит. Значит, согласно теореме Куммера (или из-за вида треугольника Серпинского) все числа $\binom{l}{m}$ нечетны, т. е. при $\|x\|\geqslant m$

$$f(x_1,\dots,x_n)=\sum_{1\leqslant i_1<\dots<i_m\leqslant n} x_{i_1}\dots x_{i_m}=\binom{\|x\|}{m}\mod 2=1,$$

а при $\|x\|<m$, очевидно, $f(x_1,\dots,x_n)=0$, так как каждое слагаемое в указанной выше сумме будет равно нулю, потому что единиц среди чисел x_i в этом случае меньше m. Отсюда следует монотонность функции $f(x_1,\dots,x_n)$.

6. Многочлены Жегалкина

Докажем следующую теорему.

Теорема 3 (Жегалкин). *Любую булеву функцию можно единственным образом представить в виде многочлена Жегалкина.*

Доказательство. Доказать возможность представления произвольной булевой функции в виде многочлена Жегалкина можно разными способами. Один из них, не самый простой для понимания, но зато самый быстрый из известных, будет изложен сразу после этой теоремы. Но все эти способы дают один и тот же результат, так как для любой булевой функции существует только один реализующий ее многочлен Жегалкина. Действительно, различных булевых функций от переменных $x_1,\dots,x_n$ имеется ровно 2^{2^n}, так как каждое из 2^n значений $f(\alpha_1,\dots,\alpha_n)$, где $\alpha_i=0,1$, $i=1,\dots,n$, из таблицы можно выбрать двумя способами, а различных многочленов Жегалкина от тех же переменных тоже имеется ровно 2^{2^n}, так как каждый из них однозначно задается набором своих 2^n двоичных коэффициентов

$c_{\alpha_1\ldots\alpha_n}$, $\alpha_i = 0, 1$, $i = 1, \ldots, n$. Если бы какие-то два многочлена реализовали одну функцию, то многочленов для реализации всех функций не хватило бы. □

Найдем явные формулы, с помощью которых можно из таблицы значений булевой функции получить строку коэффициентов ее многочлена Жегалкина. Рассмотрим вначале случай одной переменной.

Любая функция $f(x)$ представляется в виде многочлена Жегалкина как $c_0 \oplus c_1 x$, где коэффициенты c_0, c_1 — нули или единицы.

Задача 24. Проверьте, что

$$c_0 = f(0) = 1 \,\&\, f(0) \oplus 0 \,\&\, f(1),$$
$$c_1 = f(0) \oplus f(1) = 1 \,\&\, f(0) \oplus 1 \,\&\, f(1).$$

Если записать коэффициенты двух указанных выше линейных функций в виде квадратной таблицы, то она будет иметь вид

$$C_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Матрица C_1 образована коэффициентами преобразования столбца значений функции одной переменной в строку коэффициентов ее многочлена Жегалкина.

Любая функция $f(x_1, x_2)$ представляется в виде многочлена Жегалкина как

$$c_{00} \oplus c_{10} x_1 \oplus c_{01} x_2 + c_{11} x_1 x_2,$$

где коэффициенты c_{ij} — нули или единицы.

Задача 25. Проверьте, что

$$c_{00} = f(0,0), \quad c_{10} = f(0,0) \oplus f(1,0), \quad c_{01} = f(0,0) \oplus f(0,1),$$
$$c_{11} = f(0,0) \oplus f(1,0) \oplus f(0,1) \oplus f(1,1).$$

Удобно перенумеровать c_{ij} и $f(i,j)$ как c_0, c_1, c_2, c_3 и f_0, f_1, f_2, f_3, если сопоставить каждому набору (i,j) его двоичный номер

$$(i,j)_2 = i + 2j = 0, 1, 2, 3.$$

Если записать коэффициенты четырех указанных выше линейных функций в виде квадратной таблицы, то она будет иметь вид

$$C_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Матрица C_2 образована коэффициентами преобразования столбца значений функции двух переменных в строку коэффициентов ее многочлена Жегалкина.

Разберемся теперь, как выполняется такое преобразование для произвольной функции трех переменных. Столбец значений для удобства перенумеруем в соответствии с двоичной нумерацией:

$$f_0 = f(0,0,0), \quad f_1 = f(0,0,1), \quad f_2 = f(0,1,0), \quad f_3 = f(0,1,1),$$
$$f_4 = f(1,0,0), \quad f_5 = f(1,0,1), \quad f_6 = f(1,1,0), \quad f_7 = f(1,1,1).$$

Аналогично занумеруем строку коэффициентов многочлена Жегалкина:

$$c_0 = c_{000}, \quad c_1 = c_{001}, \quad c_2 = c_{010}, \quad c_3 = c_{011},$$
$$c_4 = c_{100}, \quad c_5 = c_{101}, \quad c_6 = c_{110}, \quad c_7 = c_{111}.$$

Сам многочлен $P(x_1, x_2, x_3)$ поэтому можно записать в виде

$$c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus c_3x_1x_2 \oplus c_4x_3 \oplus c_5x_1x_3 \oplus c_6x_2x_3 \oplus c_7x_1x_2x_3.$$

Разложим его по переменной x_3:

$$P(x_1, x_2, x_3) = P(x_1, x_2, 0) + x_3P(x_1, x_2, 1) = P_0(x_1, x_2) + x_3P_1(x_1, x_2).$$

Очевидно, что

$$P_0(x_1, x_2) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus c_3x_1x_2,$$
$$P_1(x_1, x_2) = c_4 \oplus c_5x_1 \oplus c_6x_2 \oplus c_7x_1x_2.$$

Так как

$$P_0(x_1, x_2) = P(x_1, x_2, 0) = f(x_1, x_2, 0),$$
$$P_0(x_1, x_2) \oplus P_1(x_1, x_2) = P(x_1, x_2, 1) = f(x_1, x_2, 1),$$

получаем

$$P_1(x_1, x_2) = f(x_1, x_2, 0) \oplus f(x_1, x_2, 1) = f_0(x_1, x_2) \oplus f_1(x_1, x_2),$$

где $f(x_1, x_2, i)$ для краткости обозначено $f_i = f_i(x_1, x_2)$. Обозначим через T преобразование, переводящее функцию f в реализующий ее многочлен Жегалкина P, а через t — соответствующее преобразование значений функции в его коэффициенты. Тогда, очевидно,

$$P_0 = T(f_0), P_1 = T(f_0 \oplus f_1),$$
$$(c_0, c_1, c_2, c_3) = t(f_0, f_1, f_2, f_3),$$
$$(c_4, c_5, c_6, c_7) = t(f_0 \oplus f_4, f_1 \oplus f_5, f_2 \oplus f_6, f_3 \oplus f_7).$$

Преобразование t состоит из 8 функций t_i от переменных f_i, $i = 0, \ldots, 7$, и каждая из них представляется многочленом Жегалкина первой

степени с нулевым свободным членом. Такие многочлены далее называются линейными (и само преобразование t тоже называется поэтому линейным). Любой из них можно записать в виде

$$t_i = c_{i0} \,\&\, f_0 \oplus \ldots \oplus c_{i7} \,\&\, f_7,$$

где коэффициенты c_{ij} — нули или единицы. Докажем это. Очевидно, что сумма по модулю два любых линейных функций будет линейной функцией, и для любой линейной функции $f(x_1, \ldots, x_n)$ справедливо тождество

$$f(x_1 \oplus y_i, \ldots, x_n \oplus y_n) = f(x_1, \ldots, x_n) \oplus f(y_1, \ldots, y_n).$$

Из этого тождества и линейности функций t_0, t_1, t_2, t_3 (см. задачу 25) с учетом равенства

$$(c_4, c_5, c_6, c_7) = t(f_0 \oplus f_4, f_1 \oplus f_5, f_2 \oplus f_6, f_3 \oplus f_7)$$

получаем, что и t_4, t_5, t_6, t_7 — линейные функции от переменных f_i, $i = 0, \ldots, 7$, причем коэффициенты у них при парах переменных t_i, t_{i+4}, $i = 0, 1, 2, 3$, одинаковые и совпадают с коэффициентами при переменных f_i, $i = 0, 1, 2, 3$, у функций t_0, t_1, t_2, t_3 соответственно. Отсюда можно сделать два вывода.

Во-первых, таблица коэффициентов c_{ij}, $i, j = 0, \ldots, 7$, линейных функций t_i, $i = 0, \ldots, 7$, — это квадрат размера 8×8, который разбивается средними линиями на 4 квадрата размера 4×4, один из которых нулевой (состоит из одних нулей), а три остальных — одинаковые и совпадают с матрицей C_2:

$$C_3 = \begin{pmatrix} C_2 & 0 \\ C_2 & C_2 \end{pmatrix} = \left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right).$$

Аналогичным образом можно получить таблицу C_4 размера 16×16 и вообще таблицу C_n размера $2^n \times 2^n$ коэффициентов линейного преобразования $t(f_0, \ldots, f_{2^n-1})$:

$$C_n = \begin{pmatrix} C_{n-1} & 0 \\ C_{n-1} & C_{n-1} \end{pmatrix}.$$

Непосредственно видно, что таблицы C_1, C_2, C_3, C_4 совпадают с уже знакомыми нам «таблицами Паскаля по модулю два», которые ино-

гда называют также матрицами Серпинского. По индукции можно доказать, что так будет и для любой таблицы C_n.

Во-вторых, можно заметить, что если обозначить через $L(n)$ число операций сложения по модулю два, используемых в указанном выше алгоритме для вычисления по значениям булевой функции n переменных коэффициентов ее многочлена Жегалкина, то $L(3) = 2L(2) + 4$, так как для вычисления

$$(c_0, c_1, c_2, c_3) = t(f_0, f_1, f_2, f_3)$$

нужно $L(2)$ операций, потом 4 операции для вычисления

$$f_0 \oplus f_4, \quad f_1 \oplus f_5, \quad f_2 \oplus f_6, \quad f_3 \oplus f_7$$

и еще $L(2)$ операции для вычисления

$$(c_4, c_5, c_6, c_7) = t(f_0 \oplus f_4,\ f_1 \oplus f_5,\ f_2 \oplus f_6,\ f_3 \oplus f_7).$$

Аналогично получается равенство $L(n) = 2L(n-1) + 2^{n-1}$. Очевидно, $L(1) = 1$.

Задача 26. Докажите по индукции, что $L(n) = 2^{n-1}n$.

Таким образом, для вычисления коэффициентов многочлена Жегалкина для данной булевой n-местной функции достаточно $2^{n-1}n$ битовых операций сложения по модулю два. Так как для этого вычисления надо использовать все 2^n значений данной функции, ясно, что указанный алгоритм если не оптимальный, то не очень далек от оптимального. Является ли он оптимальным, до сих пор неизвестно.

Задача 27. Докажите, что преобразование, обратное к преобразованию $c_i = t_i(f_0, \ldots, f_{2^n-1})$, то есть преобразование, вычисляющее по коэффициентам $c_0, \ldots, c_{2^n-1}$ многочлена Жегалкина от n переменных значения $f_0, \ldots, f_{2^n-1}$ реализуемой им функции, совпадает с исходным преобразованием.

И наконец, укажем одно применение рассмотренного алгоритма[12]. Его можно использовать для того, чтобы быстро умножать многочлены Жегалкина. Действительно, пусть даны два многочлена P_1, P_2 от n переменных. Нужно найти их произведение P_1P_2 и преобразовать его тоже в многочлен Жегалкина. Для этого можно перемножить каждую пару одночленов из P_1 и P_2, в полученном одночлене устра-

[12] Об этом автор узнал от профессора кафедры дискретной математики мехмата МГУ А. В. Чашкина.

нить квадраты, так как $x^2 = x$, а потом привести подобные члены, уничтожив одинаковые пары многочленов. Таким образом, получим многочлен, в котором не будет высоких степеней переменных и все одночлены будут различны, т. е. многочлен Жегалкина. Но число операций в подобном алгоритме будет не меньше 2^{2n}. Оказывается, есть другой простой алгоритм, выполняющий указанное умножение за $3 \cdot 2^{n-1}n + 2^n$ операций $\oplus$ и &. В нем нужно сначала со сложностью $2^n n$ по коэффициентам многочленов P_1, P_2 вычислить таблицу значений реализуемых ими булевых функций, потом, выполнив 2^n операций &, вычислить таблицу значений функции $P_1 \,\&\, P_2$ и с помощью $2^{n-1}n$ операций $\oplus$ восстановить по ним коэффициенты многочлена Жегалкина, являющегося произведением P_1 и P_2.

Заключительные разделы содержат некоторые дополнения к основной теме книжки.

7. Кое-что о записи чисел в двоичной системе

Для любого n обозначим через $\lambda(n)$ уменьшенную на единицу длину двоичной записи числа n, а $\nu(n)$ — ее сумму цифр (другими словами, число единиц в ней).

Очевидно, что $\lambda(n) + \nu(n) - 1 \leqslant 2\lambda(n)$. Те, кто знают логарифмы, сообразят, что $\lambda(n) = \lfloor \log_2 n \rfloor$, где знак $\lfloor x \rfloor$ означает целую часть числа x. Но можно вычислить обе введенные функции, даже не упоминая о двоичной записи. Для этого надо воспользоваться следующими правилами:

$$\nu(1) = 1, \quad \nu(2n) = \nu(n), \quad \nu(2n+1) = \nu(n) + 1,$$
$$\lambda(1) = 0, \quad \lambda(2n) = \lambda(2n+1) = \lambda(n) + 1.$$

Однако для доказательства справедливости этих правил полезно, конечно, воспользоваться двоичной системой, после чего они становятся почти очевидными. Заметим, что последнее правило позволяет рекуррентно вычислять $\lambda(n)$ и тем самым дает определение $\lambda(n)$, независимое от понятия двоичного логарифма (и, значит, дает представление о двоичном логарифме для тех, кто не знает логарифмов).

Задача 28. Докажите неравенство $\nu(n+1) \leqslant \nu(n) + 1$.

Указание. Оно, очевидно, превращается в равенство, если n четно, так как тогда его двоичная запись заканчивается нулем. Если же эта двоичная запись заканчивается k единицами, перед которыми

стоит нуль, то двоичная запись числа $n+1$ заканчивается k нулями, перед которыми стоит единица (а старшие биты остаются без изменения, если они есть). Для того, чтобы в этом убедиться, достаточно выполнить прибавление 1 к n в двоичной системе. В обоих рассмотренных случаях $\nu(n+1) \leqslant \nu(n)+1$.

Задача 29. Докажите неравенство $\lambda(n+1)+\nu(n+1)\leqslant\lambda(n)+\nu(n)+1$.

Указание. Действительно, если $2^{k-1} < n+1 < 2^k$, то $\lambda(n+1) = k-1 = \lambda(n)$, и из неравенства $\nu(n+1) \leqslant \nu(n)+1$ следует нужная оценка. Если же $n+1=2^k$, то $\lambda(n+1)=k=\lambda(n)+1$, $\nu(n+1)=1$, $\nu(n)=k$, откуда следует, что $\lambda(n+1)+\nu(n+1)=k+1\leqslant 2k=\lambda(n)+\nu(n)+1$.

Задача 30. Докажите равенство $\lambda(2n)+\nu(2n)=\lambda(n)+\nu(n)+1$.

Указание. Оно следует из равенств $\nu(2n)=\nu(n)$, $\lambda(2n)=\lambda(n)+1$.

8. Что такое булевы схемы

Для выполнения логических операций в электронике было создано множество устройств. Современные устройства, называемые логическими ячейками или элементами, имеют микроскопические размеры и представляют из себя особые участки кремниевого кристалла. Физика и химия протекающих в них процессов весьма сложна и не будет обсуждаться в этой книжке. Также сложна и технология их производства. Но логика работы этих элементов довольно проста: каждый из них реализует определенную логическую (или, как говорят, булеву) операцию. Элемент, реализующий (или вычисляющий) конъюнкцию, называется *конъюнктором*, на схемах иногда обозначается символом AND (рис. 4). Элемент дизъюнкции (*дизъюнктор*) на схемах иногда обозначается OR (рис. 5). Элемент, реализующий сумму по модулю два, на схемах обозначается XOR (сокращение от английского EXCLUSIVE OR — «исключающее или»; рис. 6). Для выполнения сложения однобитных чисел делают обычно даже специальный логический элемент с двумя входами x, y и двумя выходами w, v, как бы составленный из

Рис. 4. Элемент конъюнкции

Рис. 5. Элемент дизъюнкции

элемента умножения (конъюнкции) и элемента сложения по модулю два. Этот элемент часто называют *полусумматором* (рис. 7).

Рис. 6. Элемент суммы по модулю два («исключающее или»)

Мы не будем давать строгого определения понятия логической (или булевой) схемы, а всего лишь рассмотрим в качестве примера два варианта схемной реализации полусумматора — использующий элемент XOR и обходящийся без него. В первом варианте схема состоит из двух элементов, а во втором — из четырех. Число элементов в схеме далее будем называть ее *сложностью*. Сложность данной схемы S обычно обозначается $L(S)$. Сложностью схемы в значительной степени определяются ее физические размеры, т. е. площадь, занимаемая схемой на кремниевом кристалле. Как правило, чем больше сложность, тем больше площадь.

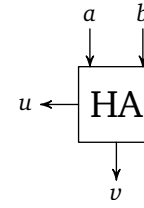


Рис. 7. Полусумматор

Другой важной характеристикой схемы является ее глубина. *Глубиной* схемы называется максимальное число ее элементов, образующих цепь, соединяющую какой-либо вход схемы с одним из ее выходов. Например, у схемы на рис. 8 глубина равна единице, а у схемы на рис. 13 — трем. Глубина схемы S обычно обозначается $D(S)$.

Глубина схемы в значительной степени определяет ее задержку. *Задержкой* схемы называется время, прошедшее от момента появления сигнала на входах схемы (как правило, значения входов схемы стабилизируются в разные момент времени, и тогда отсчет начинается с последнего из них) до момента появления сигнала на ее выходе. Как правило, чем больше глубина, тем больше задержка.

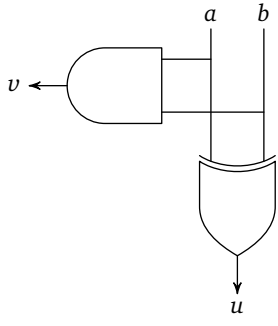


Рис. 8. Реализация полусумматора с помощью элемента XOR

Пример схемы из элементов AND и XOR, реализующей булев оператор HA (от английского Half Adder), приведен на рис. 8. Схему из элементов AND, OR, NOT (конъюнкций, дизъюнкций и отрицаний) для полусумматора можно построить, используя следую-

щую формулу: $x \oplus y = (\neg x \,\&\, y) \vee (x \,\&\, \neg y)$. Предлагается читателю нарисовать схему самостоятельно. В ней будет шесть элементов.

Задача 31. Проверьте справедливость тождества (проверить его гораздо проще, чем до него догадаться!)

$$x \oplus y = \neg(x \,\&\, y) \,\&\, (x \vee y).$$

Задача 32. Используя это тождество, постройте схему для полусумматора из четырех элементов.

9. Схемная реализация сумматора однобитных чисел

Сложность доказанной в предыдущих разделах формулы

$$f_3(x_1, \ldots, x_{15}) = \sum_{1 \leqslant i_1 < \ldots < i_8 \leqslant 15} x_{i_1} \ldots x_{i_8} = \\ = x_1 \ldots x_8 \oplus x_1 \ldots x_7 x_9 \oplus \ldots \oplus x_8 \ldots x_{15},$$

вычисляющей старший разряд суммы 15 битов, равна

$$8 \cdot \binom{15}{7} - 1 = 51479.$$

Задача 33. Проверьте это.

Указание. В многочлене Жегалкина $f_3(x_1, \ldots, x_{15})$ содержится ровно $\binom{15}{8} = \binom{15}{7}$ слагаемых. Поэтому общее число символов переменных в этой формуле равно $8 \cdot \binom{15}{7}$. Число элементов AND и XOR (символов операций $\&, \oplus$) в подобных формулах всегда на единицу меньше, чем число символов переменных.

Хотя многочлены f_0, f_1, f_2 для остальных разрядов суммы 15 битов существенно проще, вычисленная сложность очень велика и сравнима с размерами таблицы для всех четырех булевых функций, реализуемых этими многочленами. В общем случае суммы n битов сложность многочлена Жегалкина для старшего разряда этой суммы будет по порядку равна $2^n\sqrt{n}$. Это очень много. Кажется, что использование формул алгебры логики все-таки не дает существенного выигрыша в сравнении с простым применением таблиц. Но можно, оказывается, построить формулу, вычисляющую этот многочлен и содержащую не более $Cn^{3{,}03}$ символов переменных, где C — некоторая (довольно большая) константа. Эту рекордную формулу недавно придумал молодой московский математик Игорь Сергеев, научный

сотрудник кафедры дискретной математики мехмата МГУ. Привести ее построение здесь мы не можем, так как оно весьма сложно.

Но если для реализации упомянутых многочленов использовать не формулы, а булевы схемы (в которых, в отличие от формул, выходы элементов можно использовать многократно для присоединения их к входам других элементов), то сложность реализации всей схемы, вычисляющей сумму n битов, оказывается еще меньше. Такие схемы были давно известны, и построить их (в отличие от формул) несложно. Далее будет показано, как это сделать. Для краткости обозначаем эти схемы символом FA_n (от английского Full Adder).

Для примера построим схему FA_9. Представим ее в виде нескольких рисунков. На них изображены модули (подсхемы) этой схемы и на последнем — сама схема. Проверить, что эта схема работает правильно, читателю предлагается самостоятельно. Вам в этом помогут подписи к рис. 9–12 (для разных схем используются обозначения с разными верхними индексами).

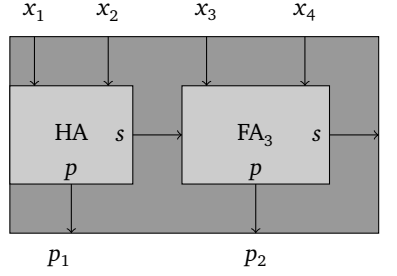


Рис. 9. Модуль FA_4^1. Входы и выходы схемы связаны соотношением $x_1 + x_2 + x_3 + x_4 = 2(p_1 + p_2) + s$

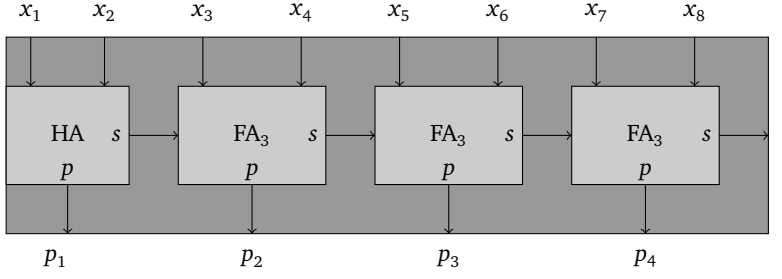


Рис. 10. Модуль FA_8^1. Входы и выходы схемы связаны соотношением $x_1 + \ldots + x_8 = 2(p_1 + \ldots + p_4) + s$

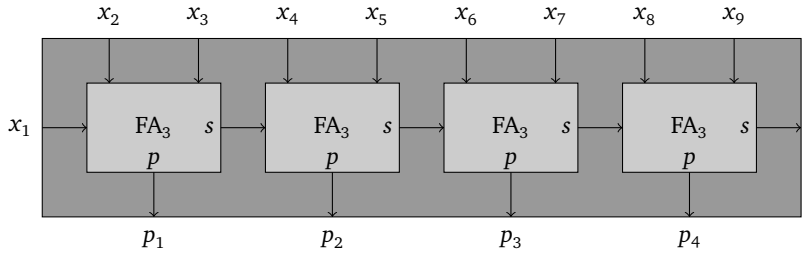


Рис. 11. Модуль FA_9^1. Входы и выходы схемы связаны соотношением $x_1 + \ldots + x_9 = 2(p_1 + \ldots + p_4) + s$

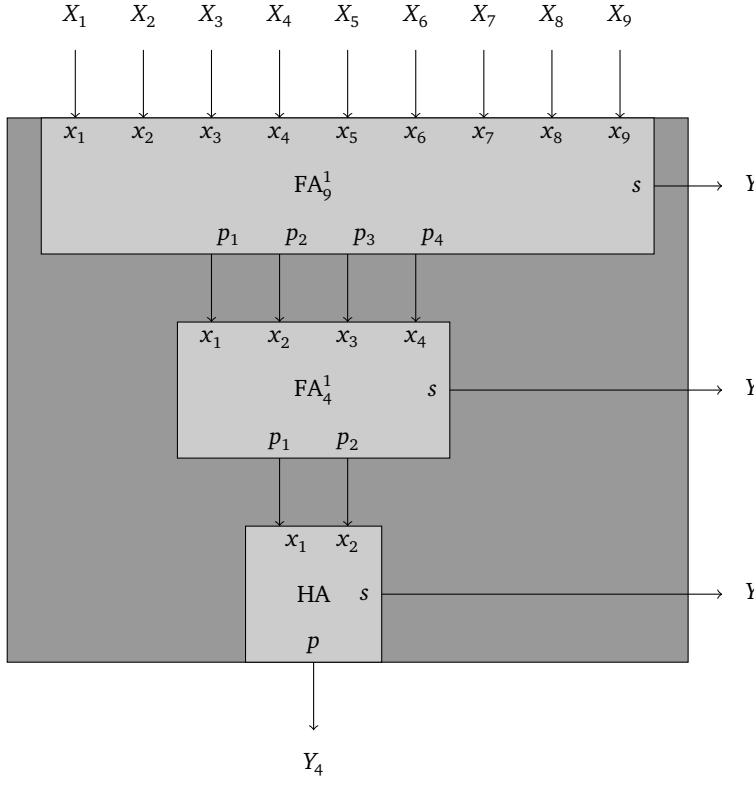


Рис. 12. Модуль FA_9. Входы и выходы схемы связаны соотношением $X_1 + \ldots + X_9 = 2^3 Y_4 + 2^2 Y_3 + 2Y_2 + Y_1$, то есть схема является счетчиком числа единиц в наборе $(X_1, \ldots, X_9)$

Задача 34. Схемы для FA_2 и FA_3 нарисуйте самостоятельно (одну из них можно увидеть на рис. 13).

Аналогично можно построить схемы и для FA_{15} (сумматора 15 битов) и вообще для сумматора n битов.

Задача 35. Докажите, что сложность схемы для сумматора n битов менее $5n$.

Недавно молодые петербургские математики А. С. Куликов и Е. А. Деменков (Санкт-Петербургское отделение МИРАН) построили схему для сумматора n битов, сложность которой асимптотически равна $4{,}5n$, а И. С. Сергеев построил еще лучшую схему, которая при почти такой же сложности имеет рекордно малую глубину, асимптотически равную $3{,}34\log_2 n$. Используя эту схему, он также построил схему для умножения n-битных чисел, сложность которой асимптотически равна $5{,}5n^2$, а глубина $4{,}34\log_2 n$. Эта глубина является в настоящий момент рекордной, но схемы для умножения с существенно большей глубиной могут иметь меньшую сложность (см. об этом, например, [1]).

10. Сумматор — схема для сложения двух двоичных чисел

Два n-разрядных двоичных числа $x = (x_n \dots x_1)_2$ и $y = (y_n \dots y_1)_2$ складываются школьным методом в столбик следующим образом:

$$\begin{array}{rr} & q_{n+1}q_n \dots q_1 \\ + & \phantom{q_{n+1}}x_n \dots x_1 \\ & \phantom{q_{n+1}}y_n \dots y_1 \\ \hline & z_{n+1}z_n \dots z_1. \end{array}$$

Числа $q_1, \dots, q_{n+1}$ — результаты переносов. Первый настоящий перенос q_2 возникает только тогда, когда $x_1 = y_1 = 1$. Каждый следующий перенос q_{i+1} возникает только тогда, когда суммируемые разряды x_i, y_i и перенос q_i из предыдущего разряда в сумме дают число не менее двух, т. е. среди этих трех чисел не меньше двух единиц. Только в этом случае $q_{i+1} = 1$, в противном случае $q_{i+1} = 0$. Булева функция от трех переменных x, y, z, которая обращается в 1, только когда $x + y + z \geqslant 2$, является функцией голосования комитета из трех человек. Ее называют также мажоритарной функцией, или медианой. Ее можно выразить через знакомые нам булевы функции двух переменных формулой $m(x, y, z) = xy \vee xz \vee yz$ или $xy \oplus xz \oplus yz$. Обе формулы

правильно вычисляют (или, как еще говорят, реализуют) медиану $m(x, y, z)$, в чем легко убедится непосредственной проверкой. Если $x = y = z = 1$, то обе формулы дают в результате $1 = m(1, 1, 1)$ (действительно, $1 \vee 1 \vee 1 = 1 = 1 \oplus 1 \oplus 1$). Если же, например, $x = y = 1$, $z = 0$, то в обоих формулах ровно одно слагаемое из трех равно единице, значит, и в этом случае обе формулы вычисляют $1 = m(1, 1, 0)$. Два оставшихся случая рассматриваются точно так же, причем в силу симметричности обеих формул и функции m ничего нового при их рассмотрении не возникнет. Пользуясь полученными формулами, легко заметить, что сложение трех однобитных чисел x, y, z можно выполнить, пользуясь формулами:

$$x + y + z = 2u + v, \quad v = x \oplus y \oplus z, \quad u = m(x, y, z) = xy \oplus yz \oplus xz.$$

Из сказанного ясно, что сложение двух n-разрядных чисел выполняется по следующим формулам (в которых $i = 1, 2, \ldots, n$):

$$\begin{cases} q_1 = 0, \\ z_i = x_i \oplus y_i \oplus q_i, \\ q_{i+1} = x_i y_i \oplus (x_i \oplus y_i) q_i, \\ z_{n+1} = q_{n+1}. \end{cases}$$

Обозначим через B_i, где $i > 1$, изображенную на рис. 13 схему сложности пять и глубины три.

Нужная нам схема A_n — сумматор двух n-разрядных двоичных чисел — получается путем последовательного соединения блоков B_i, $i = 1, \ldots, n$, как показано на рис. 14.

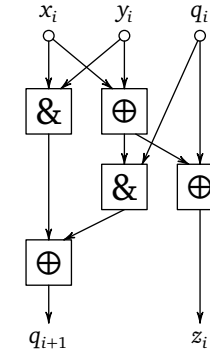


Рис. 13. Модуль вычисления очередного разряда и переноса

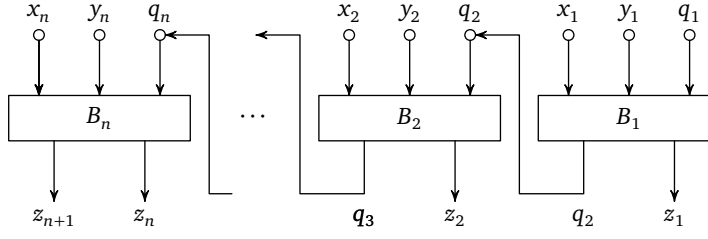


Рис. 14. Схема n-разрядного двоичного сумматора

В этой схеме блок B_1 выполняет преобразование $z_1{=}x_1{+}y_1 \pmod 2$, $q_2 = x_1 y_1$ сложности два и глубины один. Таким образом, сложность построенного сумматора $L(A_n) = 5n - 3$, а глубина $D(A_n) = 3n - 2$.

Построенная схема является оптимальной (т. е. имеет минимальную сложность), как показал около сорока лет назад Н. П. Редькин (в настоящее время профессор кафедры дискретной математики мехмата МГУ). Но глубина этой схемы минимальной не является. Задача построения сумматора минимальной глубины оказалась крайне трудной. Схему асимптотически минимальной глубины построил в конце 1960-х годов В. М. Храпченко, ведущий научный сотрудник сектора теоретической кибернетики ИПМ РАН. Недавно доцент кафедры дискретной математики мехмата МГУ М. И. Гринчук построил чуть лучшую схему, глубину которой можно оценить сверху простой формулой

$$\log_2 n + \log_2 \log_2 n + 6.$$

Наилучшая нижняя оценка глубины такой схемы (если она построена только из элементов $\&, \vee, \neg$), недавно полученная В. М. Храпченко, имеет вид

$$\log_2 n + 0{,}15 \log_2(\log_2 \log_2 n).$$

11. Подсчет числа единиц в булевой строке компьютером

Встречаются ситуации, когда компьютеру приходится выполнять операции не с числами, а с битами, составляющими эти числа (или машинные слова). Для этого используются команды языка ассемблера. Но и язык C (и C++) имеет средства для непосредственной работы с битами. Для быстрого выполнения различных манипуляций с битами в программистском фольклоре и в литературе известно множество различных эффективных программ. Замечательная коллекция таких трюков содержится в [9] (английский оригинал называется «Hacker's Delight»).

Рассмотрим задачу подсчета числа единичных битов в машинном слове (т. е. количества единичных разрядов в данном 32-битном числе x). По существу, это та же задача, с которой мы имели дело в предыдущих разделах, но решать ее здесь мы будем не с помощью булевых формул и схем, а с помощью компьютерных команд и программ. Известно много элегантных и эффективных решений этой задачи. Приведем одно из лучших.

Заметим, что рассматриваемая задача очень близка к задаче построения логической схемы, выполняющей суммирование n однобитных чисел в двоичной системе счисления. Известно, что такую схему можно построить следующим образом. Разобьем эти числа на пары и сложим каждую пару. Получим $n/2$ двухбитных чисел. Эти числа опять разобьем на пары и сложим каждую пару. Получим $n/4$ трехбитных чисел. Разобьем эти числа на пары и сложим каждую пару. Получим $n/8$ четырехбитных чисел. Далее получаем $n/16$ пятибитных чисел и т. д. В результате получится одно $(\log_2 n + 1)$-битное число (предполагаем, что n есть степень двойки), которое и равно сумме всех n чисел, а другими словами — количеству единиц среди них.

Указанную схему при $n = 32$ можно промоделировать следующей компьютерной программой. Сначала вычисляем $x \,\&\, (010101\ldots0101)_2$ и получаем число $(0x_{31}0x_{29}\ldots0x_1)_2$. Аналогично получаем число $(0x_{32}0x_{30}\ldots0x_2)_2$ в результате операции

$$x \leftarrow (x \gg 1) \,\&\, (010101\ldots0101)_2,$$

где $(x \gg 1)$ означает сдвиг битов числа x на один разряд вправо. Потом складываем полученные числа. Все вместе это выполняется следующим образом:

$$x \leftarrow x \,\&\, (010101\ldots0101)_2 + (x \gg 1) \,\&\, (010101\ldots0101)_2.$$

В результате получится число $(y_{32}\ldots y_1)_2$, для которого

$$(y_{2i}y_{2i-1})_2 = x_{2i} + x_{2i-1}, \quad i = 1, \ldots, 16.$$

Значит, выполнив 4 команды, мы параллельно сложили 16 пар чисел. Далее выполняем команды:

$$x \leftarrow (x \,\&\, (00110011\ldots0011)_2) + ((x \gg 2) \,\&\, (00110011\ldots0011)_2).$$

Результат состоит из четверок битов, которые являются двоичными записями восьми сумм

$$x_{4i} + x_{4i-1} + x_{4i-2} + x_{4i-3}, \quad i = 1, \ldots, 8.$$

Заметим, что первый бит в каждой четверке нулевой, так как сумма четырех однобитных чисел трехбитна. Далее выполняем аналогичным образом команды:

$$x \leftarrow (x \,\&\, (00001111\ldots00001111)_2) + \\ + ((x \gg 4) \,\&\, (00001111\ldots00001111)_2).$$

Результат состоит из восьмерок битов, которые являются двоичными записями четырех сумм

$$x_{8i} + x_{8i-1} + \ldots + x_{8i-7}, \quad i = 1, \ldots, 4.$$

Заметим, что левая половина битов в каждой восьмерке нулевая, так как сумма восьми однобитных чисел четырехбитна. Далее выполняем аналогичным образом команды:

$$x \leftarrow (x \,\&\, (00000000111111110000000011111111)_2) + \\ + ((x \gg 8) \,\&\, (00000000111111110000000011111111)_2).$$

В результате получится число, состоящее из двух блоков по 16 бит, которые являются двоичными записями двух сумм

$$x_{16i} + x_{16i-1} + \ldots + x_{16i-15}, \quad i = 1, 2.$$

Заметим, что левая половина битов в каждом блоке нулевая, так как сумма 16 однобитных чисел пятибитна. Далее выполняем аналогичным образом команды:

$$x \leftarrow (x \,\&\, (00000000000000001111111111111111)_2) + \\ + ((x \gg 16) \,\&\, (00000000000000001111111111111111)_2)$$

и получаем окончательно нужную нам сумму, в которой левая половина битов нулевая, а ненулевыми могут быть только правые 6 битов. Заметим, что каждую строку, кроме первой и второй, можно упростить, устраняя первую конъюнкцию и перегруппировывая скобки. Например, третью скобку можно переписать в виде

$$x \leftarrow (x + (x \gg 4)) \,\&\, (00001111 \ldots 00001111)_2.$$

При этом результат работы каждой строки не изменится, так как каждый из четырехбитных блоков, на которые разбивается число x, полученное в результате работы первых двух строк, имеет нуль в самом левом бите, поэтому при сложении этих четырехбитных блоков не происходит переноса в другие блоки. Первую строку можно упростить, заменив на следующую:

$$x \leftarrow x - ((x \gg 1) \,\&\, (0101 \ldots 01)_2).$$

Результат при этом не изменится, потому что при вычитании в каждой паре соседних битов будет производиться операция

$$2^{2i}(2x_{2i+1} + x_{2i}) - x_{2i+1}2^{2i} = 2^{2i}(x_{2i+1} + x_{2i}).$$

Далее, эту программу можно несколько упростить, заменив последнюю строку на следующую:

$$x \leftarrow x + (x \gg 16).$$

Полученный результат будет совпадать с правильным только в 16 правых битах, но нужная нам информация содержится именно в них, точнее, даже в 6 самых правых. Поэтому для правильной работы программы достаточно добавить строку

$$\text{return } x \,\&\, (00\ldots0111111)_2.$$

Аналогичным образом можно заменить и предпоследнюю строку в исходной программе на

$$x \leftarrow x + (x \gg 8).$$

Результат тогда будет совпадать с правильным в самой правой восьмерке битов, в следующей восьмерке он, возможно, будет неправильным (ненулевым), а в следующей — опять правильным, причем сумма этих правильных результатов (чисел, двоичные записи которых задают указанные восьмерки) будет окончательным результатом программы. Поэтому если выполнять уже измененные строки

$$x \leftarrow x + (x \gg 16), \quad \text{return } x \,\&\, (00\ldots0111111)_2,$$

окончательный результат все равно будет верным. Известно также много других эффективных программ для решения той же задачи. Некоторые из них основаны на следующих формулах, которые предлагаются в виде задач.

Задача 36. Обозначим сумму битов n-битного числа x через $\|x\|$. Докажите, что

$$\|x\| = x - \left\lfloor \frac{x}{2} \right\rfloor - \left\lfloor \frac{x}{4} \right\rfloor - \ldots - \left\lfloor \frac{x}{2^{n-1}} \right\rfloor.$$

Задача 37. Обозначим $(x \ll i)^{\text{rot}}$ циклический битовый сдвиг на i позиций. Докажите, что

$$\|x\| = -\sum_{i=0}^{n-1} (x \ll i)^{\text{rot}} \mod 2^n.$$

Указание. Каждый бит при сдвигах пробегает все возможные позиции и сумма этих чисел по модулю 2^n равна $x(11\ldots1)_2 \bmod 2^n = -x$.

В качестве применения быстрого алгоритма вычисления $\|x\|$ можно быстро вычислить знакопеременную сумму битов

$$s(x) = x_1 - x_2 + x_3 - \ldots + x_{31} - x_{32}.$$

Очевидно,

$$s(x) = \|x \,\&\, (0101\ldots01)\| - \|x \,\&\, (1010\ldots10)\|, \quad -16 \leqslant s(x) \leqslant 16.$$

Остаток от деления неотрицательного числа x на 3 можно найти, не выполняя деления, вычислив $s(x)$ и заметив, что $x \bmod 3 = s(x) \bmod 3$. Далее быстрее всего воспользоваться предвычисленной таблицей остатков по модулю три для чисел от -16 до 16. В случае знаковых чисел в определении $s(x)$ нужно заменить $-x_{32}$ на $+x_{32}$. Тогда в формулу для вычисления $s(x)$ надо добавить слагаемое $((x \gg 31) \ll 1)$, если компьютер выполняет сдвиг знаковых чисел вправо с заполнением слева нулями, а если такой команды нет, но есть команда правого знакового сдвига, при котором слева все биты заполняются знаковым битом, тогда надо вычесть это слагаемое.

Для вычисления остатка от деления x на 3 можно также положить

$$s(x) = x_1 + 2x_2 + x_3 + 2x_4 + \ldots + x_{31} + 2x_{32}$$

(для положительных чисел). Эта формула вычисляется чуть быстрее:

$$s(x) = \|x\| + \|x \,\&\, (10\ldots10)_2\|.$$

Но размеры используемой далее таблицы немного возрастут.

Подобный же прием можно использовать и при вычислении остатка от деления на 7, но в нем уже придется три раза применять функцию $\|x\|$. Возможно, в этом случае более быстрой окажется программа, вычисляющая

$$s(x) = x_1 + 2x_2 + 4x_3 + x_4 + 2x_5 + 4x_6 + \ldots$$

по формуле

$$s(x) = (x \gg 30) \,\&\, (0\ldots0111)_2 + \ldots \\ \ldots + (x \gg 3) \,\&\, (0\ldots0111)_2 + x \,\&\, (0\ldots0111)_2.$$

Остаток от деления на 15 можно вычислять с помощью функции

$$s(x) = x_1 + 2x_2 + 4x_3 + 8x_4 + x_5 + 2x_6 + 4x_7 + 8x_8 + \ldots,$$

задаваемой формулой

$$s(x) = (x \gg 28) \,\&\, (0\ldots01111)_2 + \ldots \\ \ldots + (x \gg 4) \,\&\, (0\ldots01111)_2 + x \,\&\, (0\ldots01111)_2,$$

еще быстрее. Правда, размер используемой в конце таблицы возрастет до 120 чисел. Если заменить составляющие ее числа от нуля до 15

на их остатки по модулю три или пять, получим быстрые программы вычисления $x \bmod 3$ и $x \bmod 5$. Возможно, первая из них будет быстрее, чем указанная выше.

Литература

1. *Гашков С. Б.* Системы счисления и их применение. М.: МЦНМО, 2012.
2. *Гашков С. Б.* Занимательная компьютерная арифметика. Математика и искусство счета на компьютерах и без них. М.: Книжный дом «Либроком»/URSS, 2012.
3. *Гашков С. Б.* Занимательная компьютерная арифметика. Быстрые алгоритмы вычислений с числами и многочленами. М.: Книжный дом «Либроком»/URSS, 2012.
4. *Гашков С. Б.* Современная элементарная алгебра в задачах и упражнениях. М.: МЦНМО, 2006.
5. *Виленкин Н. Я.* Рассказы о множествах. М.: МЦНМО, 2013.
6. *Леман А. А.* Московские математические олимпиады. М.: Просвещение, 1965.
7. Квант № 4 за 2012 г. (Задачи LXXV Моск. матем. олимпиады, 11 класс, второй день).
8. *Гашков С. Б.*, *Чубариков В. Н.* Арифметика, алгоритмы, сложность вычислений. М.: Дрофа, 2005.
9. *Уоррен Г.* Алгоритмические трюки для программистов. 2-е изд, расширенное. М.: Вильямс, 2014.

Оглавление

Гашков Сергей Борисович

Сложение однобитных чисел

Треугольник Паскаля, салфетка Серпинского
и теорема Куммера

Подписано в печать 13.03.2014 г. Формат $60\times90\,{}^{1}\!/_{16}$. Бумага офсетная.
Печать офсетная. Печ. л. 2,5. Тираж 1000 экз. Заказ №

Издательство Московского центра
непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11. Тел. (499) 241–74–83.

Отпечатано в ППП «Типография „Наука“».
121099, Москва, Шубинский пер., 6.

УДК 511.2
ББК 22.131
Г24

Гашков С. Б.

Г24 Сложение однобитных чисел. Треугольник Паскаля, салфетка Серпинского и теорема Куммера. — М.: МЦНМО, 2014. — 40 с.

ISBN 978-5-4439-0145-9

В книге рассказывается о любопытной связи задачи о сложении чисел в двоичной записи с алгеброй логики, многочленами Жегалкина, треугольником Паскаля, салфеткой Серпинского и теоремой Куммера о делимости биномиальных коэффициентов. Все необходимое для понимания разъясняется. Брошюра является расширенным вариантом лекции, прочитанной на Малом мехмате в МГУ им. Ломоносова 6 апреля 2013 г.

ББК 22.131

ISBN 978-5-4439-0145-9

Памяти моих родителей

1. Так ли просто сложение?

Всем известно, что самая простая часть школьной математики — это арифметика. А в ней самая простая операция — сложение. Но слово *сложность* родственно, очевидно, слову *сложение*. Странно, не правда ли? Что может быть сложного в сложении?

Мы постараемся далее показать, что сложение, даже в самой простой позиционной системе — двоичной — не так уж и просто, как это кажется на первый взгляд. Совсем непростой оказывается даже, казалось бы, простейшая задача — сложение одноразрядных (или, как говорят в программировании, однобитных) чисел[1].

Сформулируем эту задачу в общем виде.

Задача. Нужно сложить n чисел $x_i = 0$ или 1 и результат получить в двоичной записи, т. е. найти такие двоичные цифры y_i, чтобы выполнялось равенство

$$x_1 + \ldots + x_n = (y_m \ldots y_0)_2 = y_m 2^m + \ldots + 2y_1 + y_0, \quad 2^m > n \geqslant 2^{m-1}.$$

Начнем разбираться с простейших частных случаев. Тем, кто незнаком с двоичной системой, рекомендуем заглянуть в книжки [1–3]. Впрочем, все, что нужно, далее будет объяснено.

Рассмотрим вначале случай $n = 2$. Так как

$$0 + 0 = 0 = (00)_2, \quad 0 + 1 = 1 = (01)_2, \quad 1 + 1 = 2 = (10)_2,$$

результаты сложения можно задать таблицей 1.

Таблица 1. Сложение двух однобитных чисел

x_1	x_2	y_1	y_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Случай $n = 3$ чуть более сложен. И таблица сложения имеет вдвое больший размер (см. таблицу 2).

[1] Бит — это английское, давно уже ставшее международным, сокращение слов *binary digit — двоичная цифра*.

Таблица 2. Сложение трех однобитных чисел

x_1	x_2	x_3	y_1	y_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Ясно, что если, например, $n = 15$, то в качестве m можно взять 3 и соответствующая таблица будет иметь $2^{15} = 32768$ строк. А в строке будет $n + m + 1 = 19$ битов, значит, в компьютере эта таблица будет занимать не менее 76 килобайт памяти (а может, и больше, — все зависит от того, как ее в памяти хранить). Компьютер с ней справится, но вручную с такой таблицей работать невозможно. А если $n = 30$, не поможет и компьютер. Очевидно, язык таблиц для нашей задачи (и подобных ей) неудобен. Удобным языком является язык формул алгебры логики. Хотите получить представление о том, что это такое, — читайте следующий раздел.

2. Двоичная система в математике и электронике

Главное достоинство двоичной системы — простота алгоритмов арифметических операций. Таблица умножения в ней совсем не требует ничего запоминать: ведь любое число, умноженное на нуль, равно нулю, а умноженное на единицу — равно самому себе.

Таблица сложения в двоичной системе чуть сложнее таблицы умножения (в отличие от десятичной системы), потому что $1+1=10$ и возникает перенос в следующий разряд. В общем виде операцию сложения однобитовых чисел можно записать в следующем виде: $x + y = 2w + v$, где w, v — биты результата.

Внимательно посмотрев на таблицу 3, можно заметить, что бит переноса w — это просто произведение xy, потому что он равен единице, лишь когда x и y равны единице. Произведение (конъюнкция) обычно обозначается символом &. А вот бит v равен $x + y$, за исключением случая $x = y = 1$, когда он равен не 2, а 0. Операцию,

Таблица 3. Сложение в двоичной системе

x	y	w	v
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

с помощью которой по битам x, y вычисляют бит v, называют по-разному. Мы будем использовать для нее название *сложение по модулю* 2 и символ $\oplus$. Таким образом, сложение битов выполняется фактически не одной, а двумя операциями — конъюнкцией и сложением по модулю 2.

Если отвлечься от технических деталей, то именно с помощью этих операций и выполняются все операции в компьютере.

Кроме этих операций, часто используется операция *дизъюнкция*, обозначаемая далее $x \vee y$. Она отличается от операции $x \oplus y$ только тем, что $1 \vee 1 = 1$, а $1 \oplus 1 = 0$. Также полезна операция *отрицания* $\neg x = 1 - x = 1 \oplus x$. Указанные операции связаны между собой множеством тождеств.

Задача 1. Докажите тождества:

$$x \vee y = \neg(\neg x \,\&\, \neg y),$$
$$x \,\&\, y = \neg(\neg x \vee \neg y),$$
$$x \oplus y = (x \,\&\, \neg y) \vee (y \,\&\, \neg x),$$
$$x \vee y = x \oplus y \oplus (x \,\&\, y).$$

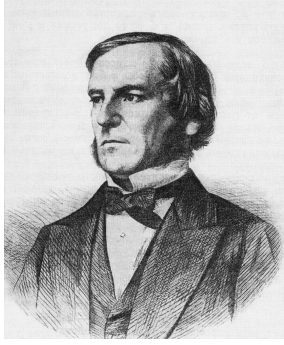
Отрицание обозначается также чертой сверху, например, вместо $\neg(x \,\&\, y \,\&\, z)$ пишут $\overline{x \,\&\, y \,\&\, z}$. Можно выразить указанные операции, называемые логическими, через обычные арифметические операции. Конъюнкция, например, просто совпадает с обычным умножением, поэтому ее при записи часто обозначают точкой или вообще пропускают.

Задача 2. Докажите тождества:

$$x \oplus y = x + y - 2xy,$$
$$x \vee y = x + y - xy,$$
$$x \vee y = \max(x, y) = 1 - \min(1 - x, 1 - y),$$
$$x \,\&\, y = \min(x, y) = 1 - \max(1 - x, 1 - y).$$

Двоичная система и логические операции

Почему эти операции называют логическими? Потому что если сопоставить каждому высказыванию A его «истинностное» значение $|A| = 0$, если A ложно, и $|A| = 1$, если A истинно, то истинностное значение составного высказывания «A и B» можно выразить[2] через истинностные значения высказываний A, B по формуле



Джордж Буль

$$|A \text{ и } B| = |A| \,\&\, |B|$$

и, аналогично,

$$|A \text{ или } B| = |A| \vee |B|.$$

В последней формуле мы предполагали, что высказывание «A или B» будет истинным и тогда, когда оба высказывания A и B истинны. Иногда союз «или» понимают в несколько другом, разделительном, смысле: составное высказывание «A или B» считается истинным только в случае, если ровно одно из высказываний A, B истинно, но не оба сразу. В этом случае для вычисления истинностного значения «A или B» можно использовать формулу

$$|A \text{ или } B| = |A| \oplus |B|.$$

3. Подсчет числа единиц в двоичной строке

Итак, нам нужно сложить n чисел $x_i = 0, 1$ и результат получить в двоичном виде, т. е.

$$x_1 + \ldots + x_n = (y_m \ldots y_0)_2, \quad 2^{m+1} > n \geqslant 2^m.$$

Как выразить y_i через $x_1, \ldots, x_n$? Очевидно, что y_i выражается через $x_1, \ldots, x_n$ однозначно, т. е. является некоторой функцией $f_i(x_1, \ldots, x_n)$ от переменных x_j. Эти переменные принимают значения нуль и единица, и сама функция тоже принимает только такие значения. Такие

[2] Впервые это сделал Джордж Буль (1815–1864) — английский, а точнее ирландский, математик, один из основателей математической логики. Был профессором университета в городе Корк и отцом шести дочерей, одна из которых стала математиком, другая — первой женщиной-профессором химии в Англии, а еще одна — писательницей, известной под именем Этель Лилиан Войнич.

Таблица 4. Булевы функции, связанные со сложением двух однобитных чисел

x_1	x_2	f_1	f_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

функции называют функциями алгебры логики или булевыми функциями (в честь Дж. Буля)[3].

Рассмотрим вначале простейший случай $n = 2$. Так как

$$0 + 0 = 0 = (00)_2, \quad 0 + 1 = 1 = (01)_2, \quad 1 + 1 = 2 = (10)_2,$$

функции f_1, f_0 от переменных x_1, x_2 можно задать таблицей, по существу совпадающей с таблицей 4.

Но более компактно вместо таблицы определить функции формулами $f_1(x_1, x_2) = x_1 \,\&\, x_2$, $f_0(x_1, x_2) = x_1 \oplus x_2$. Случай $n = 3$ чуть более сложен. В нем должно выполняться тождество

$$x_1 + x_2 + x_3 = 2y_1 + y_0, \quad y_1 = f_1(x_1, x_2, x_3), \quad y_0 = f_0(x_1, x_2, x_3)$$

для некоторых булевых функций f_0, f_1. Эти функции можно задать таблицей 5. Для того чтобы ускорить ее заполнение, можно заметить, что функции f_i симметрические, т. е. они не меняют значений при любой перестановке переменных (ведь сумма $x_1 + x_2 + x_3$ не меняется при перестановке переменных.)

Таблица 5. Булевы функции для сложения трех однобитных чисел

x_1	x_2	x_3	f_1	f_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

[3] В определенном смысле булевы функции являются самыми простыми функциями из всех, встречающихся в математике. Они существенно проще, чем, например, тригонометрические функции.

Глядя на таблицу, можно заметить, что функция f_0 обладает таким свойством: она равна единице, если в наборе (x_1, x_2, x_3) нечетное число единиц (одна или три), в противном случае она равна нулю. Другими словами, она получается из обычной суммы $x_1 + x_2 + x_3$ с помощью замены результата суммирования на остаток от его деления на два. Такую манипуляцию называют приведением по модулю два и ее результат обозначают как $x_1 + x_2 + x_3 \mod 2$.

Задача 3. Проверьте, что

$$x_1 + x_2 + x_3 \mod 2 = (x_1 \oplus x_2) \oplus x_3 = x_1 \oplus (x_2 \oplus x_3)$$

и в этих тождествах переменные можно произвольным образом переставлять. Докажите, что аналогичные тождества справедливы для любого числа n булевых переменных x_i (т. е. переменных, принимающих только значения 0 или 1).

Поэтому функции $x_1 + x_2 + \ldots + x_n \mod 2$ обозначают для краткости $x_1 \oplus \ldots \oplus x_n$ (не уточняя порядок расстановки скобок, так как он несущественен). Значит, $f_0(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$.

Из таблицы также видно, что функция f_1 обладает следующим свойством: она равна единице, если в наборе (x_1, x_2, x_3) число единиц не меньше двух, иначе она равна нулю.

Задача 4. Проверьте, что тем же свойством обладает булева функция, определяемая формулой $x_1x_2 \vee x_1x_3 \vee x_2x_3$, и тем самым

$$f_1(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3.$$

В силу симметричности этих формул достаточно проверить это равенство на наборах (000), (100), (110), (111).

Задача 5. Проверьте, что функцию f_1 можно выразить (или, как еще говорят, реализовать) и формулой $f_1 = x_1x_2 \oplus x_1x_3 \oplus x_2x_3$.

Теперь можно догадаться, каким будет ответ и в общем случае. Так как формулы довольно громоздки, мы их приведем только для случая $n = 15$. Оказывается,

$$y_0 = x_1 \oplus \ldots \oplus x_{15},$$

$$y_1 = \sum_{1 \leqslant i < j \leqslant 15} x_i x_j = x_1x_2 \oplus x_1x_3 \oplus \ldots \oplus x_1x_{15} \oplus x_2x_{15} \oplus \ldots \oplus x_{14}x_{15},$$

$$y_2 = \sum_{1 \leqslant i < j < k < l \leqslant 15} x_i x_j x_k x_l = x_1x_2x_3x_4 \oplus x_1x_2x_3x_5 \oplus \ldots \oplus x_{12}x_{13}x_{14}x_{15},$$

$$y_3 = \sum_{1 \leqslant i_1 < \ldots < i_8 \leqslant 15} x_{i_1} \ldots x_{i_8} = x_1 \ldots x_8 \oplus x_1 \ldots x_7 x_9 \oplus \ldots \oplus x_8 \ldots x_{15}.$$

Приведенные формулы представляют из себя многочлены по модулю два степеней один, два, четыре, восемь. Они являются суммами по модулю два одночленов вида $x_{i_1} \ldots x_{i_m}$, не содержащих степеней переменных выше первой. Одночлены с высокими степенями переменных для реализации булевых функций не нужны, так как для булевых переменных справедливо тождество $x^2 = x$ и, как следствие, справедливы тождества $x^m = x$ при любом $m > 1$. Многочлены такого вида называются также многочленами Жегалкина[4]. Из приведенных формул видно, что использованные в них многочлены Жегалкина являются симметрическими (и по существу совпадают с многочленами, появляющимися в общей теореме Виета).

Первую формулу доказать просто. Действительно, $y_0 = 1$ тогда и только тогда, когда в наборе $(x_1 \ldots x_{15})$ число единиц нечетно. Но тогда и сумма

$$x_1 + \ldots + x_{15} = 8y_3 + 4y_2 + 2y_1 + y_0$$

нечетна, а это возможно только при $y_0 = 1$. Остальные формулы доказать сложнее. Это можно сделать разными способами. Мы укажем один из них, который приведет нас к интересным задачам из комбинаторики.

4. Треугольник Паскаля и салфетка Серпинского

Треугольник Паскаля состоит из чисел $\binom{n}{k}$, называемых биномиальными коэффициентами. В его n-й строке стоят коэффициенты $\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}$, которые получаются после раскрытия скобок и приведения подобных членов в формуле $(1+x)^n$, а именно

$$(1+x)^n = \binom{n}{0} + \binom{n}{1} x + \ldots + \binom{n}{n} x^n.$$

Открытие этого треугольника приписывается знаменитому французскому математику, физику и религиозному философу Блезу Паскалю (хотя в том или ином виде подобные таблицы были известны

[4] В честь профессора мехмата МГУ И. И. Жегалкина (1869–1947), доказавшего в 1927 г., что любую булеву функцию можно единственным способом реализовать такими многочленами.

Блез Паскаль

и до него, например, китайцам, да и в Западной Европе до Паскаля подобную таблицу, только прямоугольную, приводил в своей книге итальянец Никколо Тарталья — тот самый, который один из первых в мире научился решать кубические уравнения). Разных задач про биномиальные коэффициенты существует великое множество, и далее мы коснемся только тех, которые нам непосредственно понадобятся. Если вы хотите еще что-нибудь прочитать про биномиальные коэффициенты, то можете взять, например, книги [4, 8].

Очевидно, что $\binom{n}{0}=1$, $\binom{n}{n}=1$. Раскрывая скобки и приравнивая коэффициенты в обеих частях тождества

$$\binom{n+1}{0}+\binom{n+1}{1}x+\ldots+\binom{n+1}{n+1}x^n=(1+x)^{n+1}=(1+x)^n(1+x)=$$
$$=\left(\binom{n}{0}+\binom{n}{1}x+\ldots+\binom{n}{n}x^n\right)(1+x),$$

получаем тождество Паскаля

$$\binom{n+1}{k+1}=\binom{n}{k+1}+\binom{n}{k},$$

которое и лежит в основе треугольника Паскаля. С его помощью можно вычислять биномиальные коэффициенты, используя только операцию сложения (см. таблицу 6).

Таблица 6. Первые 11 строк треугольника Паскаля

1										
1	1									
1	2	1								
1	3	3	1							
1	4	6	4	1						
1	5	10	10	5	1					
1	6	15	20	15	6	1				
1	7	21	35	35	21	7	1			
1	8	28	56	70	56	28	8	1		
1	9	36	84	126	126	84	36	9	1	
1	10	45	120	210	252	210	120	45	10	1

Задача 6. Последовательно используя тождество Паскаля, проверьте, что всегда выполняется равенство

$$\binom{n}{k} = \frac{n!}{k!(n-k)!},$$

где $n! = n \cdot (n-1) \ldots 2 \cdot 1$.

Задача 7. Проверьте, что $\binom{n}{k} = \binom{n}{n-k} = \frac{n(n-1)\ldots(n-k+1)}{k!}$.

Заметим, что из n скобок $(1+x)\cdots(1+x)$ выбрать k раз символ x и $n-k$ раз единицу можно в точности $\binom{n}{k}$ способами, так как

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x + \ldots + \binom{n}{n}x^n.$$

Поэтому выбрать из любых n различных символов $x_1, \ldots, x_n$ в точности k различных символов $x_{i_1}, \ldots, x_{i_k}$ тоже можно ровно $\binom{n}{k}$ способами.

Задача 8. Сколько слагаемых в многочленах Жегалкина, выражающих y_0, y_1, y_2, y_3?

Ответ: у многочлена y_i в точности $\binom{15}{2^i}$ слагаемых.

Обозначим через

$$\|x\| = x_0 + \ldots + x_{15}$$

число единиц в наборе $(x_0, \ldots, x_{15})$.

Задача 9. Докажите, что значение $y_i = f_i(x_1, \ldots, x_{15})$ равно

$$\binom{\|x\|}{2^i} \bmod 2$$

(здесь и далее удобно считать, что $\binom{n}{k} = 0$ при $k > n$).

Указание. Число единиц в сумме для y_i равно $\binom{\|x\|}{2^i}$.

Далее нас будут интересовать не сами биномиальные коэффициенты, а их значения по модулю два. Удобный способ их вычислить состоит в том, что вместо вычисления биномиальных коэффициентов и деления их на два с остатком можно построить треугольник Паскаля по модулю два так, как показано в таблице 7.

Глядя на таблицу[5] 7, легко заметить, что первый ее столбец состоит только из единиц, во втором столбце нули и единицы чередуются, в третьем столбце нули и единицы стоят парами, и эти пары

[5] Она квадратная, а не треугольная, но ее половина, лежащая выше диагонали, заполнена нулями, что соответствует принятому выше соглашению $\binom{n}{k}=0$ при $n<k$, так что фактически это тот же треугольник Паскаля, только вычисленный по модулю два.

Таблица 7. Треугольник Паскаля по модулю два

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

чередуются (т. е. четверка 0011 повторяется периодически), в пятом столбце нули и единицы группируются по четыре (т. е. повторяется периодически восьмерка битов 00001111), в девятом столбце нули и единицы группируются по восемь (т. е. повторяется периодически шестнадцать битов 0000000011111111) и так далее. Уже этого наблюдения достаточно, чтобы обосновать то, что указанные выше многочлены вычисляют в точности двоичные цифры интересующей нас суммы 15 битов[6]. Для строгого доказательства полезно обратить также внимание на тот факт, что треугольник Паскаля по модулю два размера 2^n (обозначим его T_n) состоит из трех равных треугольников T_{n-1} (треугольников Паскаля по модулю два вдвое меньшего размера) и одного треугольника, заполненного нулями, как показано на рис. 1.

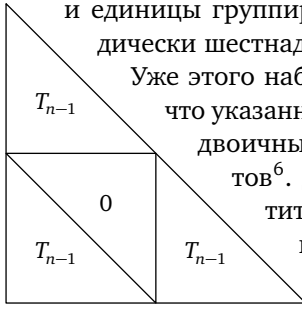


Рис. 1. Фрактальная структура треугольника Паскаля по модулю два

Для желающих строго обосновать сделанные наблюдения предлагаем решить следующие задачи.

Задача 10. Докажите, что все числа в строке с номером $2^n - 1$ треугольника Паскаля нечетны (равны единице по модулю два).

[6] Древние индусы вместо доказательств писали просто — «Смотри!».

Задача 11. Докажите, что все числа в 2^n-й строке треугольника Паскаля, кроме первого и последнего, четны.

Из обоснованной выше рекурсивной конструкции можно вывести следующее утверждение.

Задача 12. Пусть $n = (n_m \dots n_0)_2$ — двоичная запись числа n. Докажите, что

$$\binom{n}{2^k} \bmod 2 = n_k, \quad k = 0, \dots, m.$$

Это утверждение и позволяет легко доказать, что указанные многочлены Жегалкина реализуют функции f_i. Действительно, достаточно заметить, что $\|x\| = 8y_3 + 4y_2 + 2y_1 + y_0$, а значение

$$f_i(x_1, \dots, x_{15}) = \binom{\|x\|}{2^i} \bmod 2 = \binom{8y_3 + 4y_2 + 2y_1 + y_0}{2^i} \bmod 2 = y_i,$$
$$i = 0, \dots, 3.$$

Аналогично проводится доказательство и в общем случае.

Следующая задача обобщает предыдущую.

Задача 13. Пусть $n = (n_m \dots n_0)_2$ — двоичная запись числа n, а $k = (k_m \dots k_0)_2$ — двоичная запись числа k. Докажите, что $\binom{n}{k}$ нечетно тогда и только тогда, когда $k_i \leqslant n_i$ при всех $i = 0, \dots, m$.

Из предыдущей задачи легко следует

Задача 14. Докажите, что количество нечетных чисел в любой строке треугольника Паскаля равно степени двойки.

Следующая задача, как и предыдущая, нам для обоснования формул для сложения уже не нужна, но она любопытна и сама по себе.

Задача 15. Докажите, что число единиц в треугольнике Паскаля по модулю два с основанием длины 2^n равно 3^n.

Указание. Треугольник T_n состоит из трех треугольников T_{n-1} и треугольника, заполненного нулями.

Можно еще заметить, глядя на таблицу 7, что в девятом столбце верхняя половина состоит сплошь из нулей, а нижняя — сплошь из единиц. Это значит, что булева функция $f_3(x_1, \dots, x_{15}) = 1$ тогда и только тогда, когда $\|x\| = x_1 + \dots + x_{15} \geqslant 8$, значит, в формуле для нее можно заменить везде операцию $\oplus$ сложения по модулю два на операцию $\vee$ дизъюнкции и вместо многочлена Жегалкина

представить f_3 в виде

$$f_3(x_1,\ldots,x_{15}) = \bigvee_{1\leqslant i_1<\ldots<i_8\leqslant 15} x_{i_1}\ldots x_{i_8} = \\ = x_1\,\&\ldots\&\,x_8 \vee x_1\,\&\ldots\&\,x_7\,\&\,x_9 \vee \ldots \vee x_8\,\&\ldots\&\,x_{15}.$$

Приведенная выше формула является так называемой монотонной дизъюнктивной нормальной формой, а сама функция $f_3(x_1,\ldots,x_{15})$ является монотонной булевой функцией. По определению функция монотонна, если при увеличении любой ее переменной значение функции не уменьшается.

Задача 16. Проверьте, что остальные функции f_0, f_1, f_2 от переменных $x_1,\ldots,x_{15}$ не монотонны.

Разумеется, приведенные выше утверждения верны и в общем случае (см. задачу 23 на с. 20).

Треугольник Паскаля по модулю два тесно связан с так называемым треугольником Серпинского[7]. Треугольник Серпинского определяется следующим образом. Правильный треугольник разрезается средними линиями на четыре равных треугольника и средний из них удаляется. С каждым из оставшихся трех треугольников поступаем аналогично и этот процесс продолжаем до бесконечности. Те точки, которые не будут удалены из треугольника, образуют треугольник Серпинского (называемый еще иногда салфеткой Серпинского). Построение треугольника Серпинского осуществляется рекурсивным образом и проиллюстрировано на рис. 2.

Вацлав Серпинский

Окончательно салфетка Серпинского выглядит приблизительно как на рис. 3. Если на выполнение $(n+1)$-го шага рекурсии, т. е. на вырезание треугольных дырок в каждом из получившихся на предыдущем шаге 3^n треугольников, тратить $(1/2^n)$-ю долю секунды, то за две секунды все будет закончено!

[7] Вацлав Франциск Серпинский (1882–1969) — выдающийся польский математик и популяризатор науки. Учился в Варшавском университете, где тогда преподавание велось на русском языке. Дипломную работу по теории чисел защитил под руководством Г. Ф. Вороного.

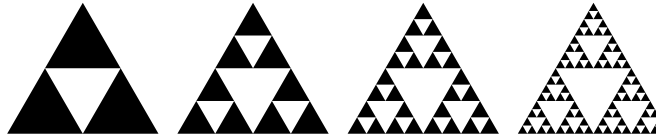


Рис. 2. Построение треугольника Серпинского.
Первые четыре шага рекурсии

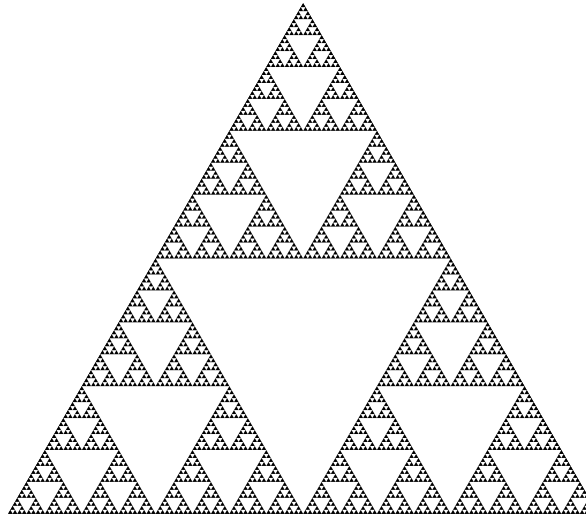


Рис. 3. Треугольник Серпинского

Серпинский придумал ее еще в 1915 г. вместе с другими подобными множествами, например, квадратом (или ковром) Серпинского, кубом Серпинского, тетраэдром Серпинского — см. об этом хорошую научно-популярную книгу [5]. Но наибольшую известность в широких кругах это множество получило на рубеже XX–XXI веков в связи с вошедшим в моду понятием фрактала. Оказалось, что салфетка и другие множества, построенные Серпинским, являются примерами так называемых фрактальных множеств. Они обладают различными любопытным свойствами. Например, площадь салфетки (точнее, мера Лебега[8] этого множества) равна нулю. Этот факт вытекает из следующего утверждения.

[8] Анри Лебег (1875–1941) — знаменитый французский математик, создатель теории меры и интеграла Лебега.

Задача 17. Докажите, что салфетка Серпинского накрывается 3^n равными треугольниками, площадь каждого из которых равна $1/4^n$ площади треугольника, из которого вырезали эту салфетку.

Салфетка Серпинского имеет дробную размерность по Хаусдорфу[9] (она равна $\log_2 3$, если кому интересно). Есть у этой салфетки и другие интересные особенности. Не будем углубляться в эти вопросы (заинтересованного читателя отошлем к многочисленным книгам про фракталы, например к книгам основателя и популяризатора этого направления Бенуа Мандельброта), а опять вернемся к элементарной комбинаторике.

5. Переносы при сложении двоичных чисел и теорема Куммера

В этом разделе мы изучим более глубоко вопрос, на какие степени двойки могут делиться биномиальные коэффициенты, и даже рассмотрим аналогичный вопрос о делимости их на степени заданного простого числа p, хотя непосредственно с задачей о сложении однобитных чисел это никак не связано.

Будем рассматривать позиционные числовые системы с произвольным основанием $b \geqslant 2$. Обозначим через $\nu_b(n)$ сумму всех b-ичных цифр в b-ичной записи числа n. Нам понадобится

Лемма 1. *Если b-ичная запись числа n оканчивается ровно m нулями, то*

$$\nu_b(n-1) + 1 - \nu_b(n) = (b-1)m,$$

и при прибавлении к числу $n-1$ единицы число m будет равно количеству произведенных во время этой операции переносов в следующий разряд (*возможно*, $m = 0$).

Доказательство. Заметим, что b-ичная запись числа $n-1$ оканчивается ровно m цифрами $b-1$. После прибавления единицы происходит m-кратный перенос в старшие разряды и получается число n, у которого все цифры, кроме $m+1$ последних, совпадают с такими же цифрами числа $n-1$, причем $(m+1)$-я от конца цифра на 1 больше такой же цифры числа $n-1$, а последние m цифр — нули. □

[9] Феликс Хаусдорф (1868–1942) — выдающийся немецкий математик. Его книга «Теория множеств» недавно была переиздана.

Рассмотрим еще одну лемму.

Эрнст Куммер

Лемма 2 (Куммер). *Количество переносов в следующий разряд при сложении чисел k и $n-k$ в b-ичной системе счисления равно*

$$s = \frac{\nu_b(n-k) + \nu_b(k) - \nu_b(n)}{b-1}.$$

Доказательство. Утверждение вытекает из леммы 1. Действительно, если переносов не происходило, то $s = 0$. Каждый перенос уменьшает один разряд в числе n на b и увеличивает следующий разряд на 1, в результате рассматриваемая величина возрастает на $b-1$, а значит, s возрастает на 1. □

Эта лемма была нужна Куммеру[10] для доказательства следующей теоремы.

Теорема 1 (Куммер). *Пусть* $\operatorname{ord}_p \binom{n}{k}$ *— показатель степени, в которой простое число p входит в разложение биномиального коэффициента*

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Тогда

$$\operatorname{ord}_p \binom{n}{k} = \frac{\nu_p(n-k) + \nu_p(k) - \nu_p(n)}{p-1},$$

где в правой части стоит количество переносов в следующий разряд при сложении чисел k и $n-k$ в p-ичной системе счисления.

Для доказательства этой теоремы полезна следующая формула Лежандра[11].

Теорема 2 (Лежандр). *Показатель степени, в которой простое число p входит в разложение факториала $n!$, равен*

$$\operatorname{ord}_p n! = \frac{n - \nu_p(n)}{p-1}.$$

[10] Эрнст Эдуард Куммер (1810–1893) — знаменитый немецкий математик, доказавший теорему Ферма во многих частных случаях и заложивший основы теории алгебраических чисел.

[11] Адриен Мари Лежандр (1752–1833) — выдающийся французский математик. Единственное дошедшее до нас изображение Лежандра — карикатура. То, что в течение долгого времени считали портретом А. М. Лежандра, является на самом деле портретом его однофамильца.

Доказательство. Действительно, если согласно предположению индукции

$$\operatorname{ord}_p(n-1)! = \frac{n-1-\nu_p(n-1)}{p-1} \quad \text{и} \quad \operatorname{ord}_p(n) = m,$$

то p-ичная запись числа n оканчивается ровно m нулями, значит, согласно лемме 1

$$\nu_p(n-1) + 1 - \nu_p(n) = (p-1)m,$$

откуда имеем

$$\operatorname{ord}_p n! = \operatorname{ord}_p(n-1)! + m = m + \frac{n-1-\nu_p(n-1)}{p-1} = \frac{n-\nu_p(n)}{p-1}.$$

База индукции $n < p$ очевидна, так как тогда $n = \nu_p(n)$. □

Задача 18. Докажите теорему Куммера.

Указание. Утверждение следует из формулы Лежандра, формулы для биномиального коэффициента и леммы 2.

С помощью теоремы Куммера легко решить задачи раздела 4, причем в более общем виде.

Задача 19. Пусть n записывается в p-ичной системе счисления при простом p в виде

$$n = (n_m \dots n_0)_p, \quad 0 \leqslant n_i < p, \quad i = 0, \dots, m = \lambda_p(n),$$

где $\lambda_p(n)$ — число, на единицу меньшее длины p-ичной записи числа n. Докажите, что тогда число биномиальных коэффициентов $\binom{n}{k}$, $0 \leqslant k \leqslant n$, не кратных p, равно $(n_0+1)\dots(n_m+1)$.

Указание. Согласно теореме Куммера число $\binom{n}{k}$ не кратно p тогда и только тогда, когда при сложении k и $n-k$ в p-ичной системе не происходит переносов в следующий разряд, т. е. когда

$$k = (k_m \dots k_0)_p, \quad 0 \leqslant k_i \leqslant n_i < p, \quad i = 0, \dots, m = \lambda_p(n).$$

Задача 20. Докажите, что n-я строка треугольника Паскаля состоит только из некратных простому p чисел тогда и только тогда, когда $n = p^m - 1$.

Еще одно решение этой задачи можно найти в [6].

Задача 21 (Московская олимпиада, 2012). Найдите число не кратных трем чисел в 2012-й строке треугольника Паскаля.

Указание. Достаточно записать число 2012 в троичной системе. Для ускорения вычислений удобно разложить его сначала в девяте-

ричной системе, а потом каждую цифру от 0 до 8 записать в троичной системе. Проще всего начать с младших разрядов. Самый младший равен остатку от деления 2012 на 9. Он, очевидно, равен 5. Отнимаем 5 и делим разность на 9. Получаем $2007 : 9 = 223$. С этим числом поступаем аналогично. Его младший разряд будет 7. После вычитания и деления на 9 получаем 24. В итоге имеем $2012 = 5 + 9(7 + 9(6 + 9 \cdot 2))$, значит, троичные цифры есть 2, 1, 1, 2, 0, 2, 2, поэтому число не кратных трем биномиальных коэффициентов равно $3 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 3 = 324$.

Задача 22 (Московская олимпиада, 2012). Докажите, что в строке треугольника Паскаля с номером 2012^{2011} количество не кратных 2011 чисел делится нацело на 2012.

Указание. Сначала надо проверить, что 2011 — простое число. Для этого достаточно установить, что оно не делится на

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43.$$

Потом надо разложить число $(p+1)^p$ в p-ичной системе при $p = 2011$. Полное разложение найти сложно, но нам достаточно найти несколько его цифр. Если n_i — эти цифры, то количество N не кратных p биномиальных коэффициентов $\binom{n}{k}$, где $n = (p+1)^p$, будет делиться на произведение чисел $(n_i + 1)$. Так как $2012 = 4 \cdot 503$, достаточно проверить, что среди цифр n_i есть хотя бы две единицы и число вида $503k - 1$, $k = 1, 2, 3$. Так как согласно формуле бинома

$$(1+p)^p = \sum_{k=0}^{p} p^k \binom{p}{k} =$$

$$= 1 + p^2 + p^3 \frac{p-1}{2} + p^4 \frac{(p-1)(p-2)}{6} + \ldots + p^{p-1} \frac{p-1}{2} + p^p + p^p$$

и все слагаемые, кроме первых трех, кратны p^4, это число при делении на p^4 равно $1 + p^2 + p^3 \frac{p-1}{2}$, поэтому младшие разряды его p-ичной записи равны $1, 0, 1, (p-1)/2 = 1005$, и этого уже достаточно, чтобы сделать вывод о делимости числа N на $(1+1)(1+1)(1005+1) = 4 \cdot 1006 = 2 \cdot 2012$. Так как сумма всех слагаемых, кроме последних двух, меньше p^p, старший разряд равен 2, поэтому N делится даже на $6 \cdot 2012$. Другое решение, в котором некоторые детали опущены, имеется в [7]. В заключение сообщим, что много задач о делимости биномиальных коэффициентов, в том числе и теоремы Куммера и Лежандра, можно найти в [8].

Задача 23. При каких n функция

$$f(x_1,\dots,x_n)=\sum_{1\leqslant i_1<\dots<i_m\leqslant n} x_{i_1}\dots x_{i_m},\quad m=\left\lceil\frac{n}{2}\right\rceil,$$

будет монотонной? Знак $\lceil x\rceil$ означает наименьшее целое число, не меньшее, чем x.

Ответ: при $n=2^k-1$, $k=1,2,3,\dots$

Указание. Если $2^{k-1}\leqslant l\leqslant 2^k-1$, то при сложении в двоичной системе чисел $m=2^{k-1}$ и $l-m<2^{k-1}$ переносов не происходит. Значит, согласно теореме Куммера (или из-за вида треугольника Серпинского) все числа $\binom{l}{m}$ нечетны, т. е. при $\|x\|\geqslant m$

$$f(x_1,\dots,x_n)=\sum_{1\leqslant i_1<\dots<i_m\leqslant n} x_{i_1}\dots x_{i_m}=\binom{\|x\|}{m}\mod 2=1,$$

а при $\|x\|<m$, очевидно, $f(x_1,\dots,x_n)=0$, так как каждое слагаемое в указанной выше сумме будет равно нулю, потому что единиц среди чисел x_i в этом случае меньше m. Отсюда следует монотонность функции $f(x_1,\dots,x_n)$.

6. Многочлены Жегалкина

Докажем следующую теорему.

Теорема 3 (Жегалкин). *Любую булеву функцию можно единственным образом представить в виде многочлена Жегалкина.*

Доказательство. Доказать возможность представления произвольной булевой функции в виде многочлена Жегалкина можно разными способами. Один из них, не самый простой для понимания, но зато самый быстрый из известных, будет изложен сразу после этой теоремы. Но все эти способы дают один и тот же результат, так как для любой булевой функции существует только один реализующий ее многочлен Жегалкина. Действительно, различных булевых функций от переменных $x_1,\dots,x_n$ имеется ровно 2^{2^n}, так как каждое из 2^n значений $f(\alpha_1,\dots,\alpha_n)$, где $\alpha_i=0,1$, $i=1,\dots,n$, из таблицы можно выбрать двумя способами, а различных многочленов Жегалкина от тех же переменных тоже имеется ровно 2^{2^n}, так как каждый из них однозначно задается набором своих 2^n двоичных коэффициентов

$c_{\alpha_1\ldots\alpha_n}$, $\alpha_i = 0, 1$, $i = 1, \ldots, n$. Если бы какие-то два многочлена реализовали одну функцию, то многочленов для реализации всех функций не хватило бы. □

Найдем явные формулы, с помощью которых можно из таблицы значений булевой функции получить строку коэффициентов ее многочлена Жегалкина. Рассмотрим вначале случай одной переменной.

Любая функция $f(x)$ представляется в виде многочлена Жегалкина как $c_0 \oplus c_1 x$, где коэффициенты c_0, c_1 — нули или единицы.

Задача 24. Проверьте, что

$$c_0 = f(0) = 1 \,\&\, f(0) \oplus 0 \,\&\, f(1),$$
$$c_1 = f(0) \oplus f(1) = 1 \,\&\, f(0) \oplus 1 \,\&\, f(1).$$

Если записать коэффициенты двух указанных выше линейных функций в виде квадратной таблицы, то она будет иметь вид

$$C_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Матрица C_1 образована коэффициентами преобразования столбца значений функции одной переменной в строку коэффициентов ее многочлена Жегалкина.

Любая функция $f(x_1, x_2)$ представляется в виде многочлена Жегалкина как

$$c_{00} \oplus c_{10} x_1 \oplus c_{01} x_2 + c_{11} x_1 x_2,$$

где коэффициенты c_{ij} — нули или единицы.

Задача 25. Проверьте, что

$$c_{00} = f(0,0), \quad c_{10} = f(0,0) \oplus f(1,0), \quad c_{01} = f(0,0) \oplus f(0,1),$$
$$c_{11} = f(0,0) \oplus f(1,0) \oplus f(0,1) \oplus f(1,1).$$

Удобно перенумеровать c_{ij} и $f(i,j)$ как c_0, c_1, c_2, c_3 и f_0, f_1, f_2, f_3, если сопоставить каждому набору (i,j) его двоичный номер

$$(i,j)_2 = i + 2j = 0, 1, 2, 3.$$

Если записать коэффициенты четырех указанных выше линейных функций в виде квадратной таблицы, то она будет иметь вид

$$C_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Матрица C_2 образована коэффициентами преобразования столбца значений функции двух переменных в строку коэффициентов ее многочлена Жегалкина.

Разберемся теперь, как выполняется такое преобразование для произвольной функции трех переменных. Столбец значений для удобства перенумеруем в соответствии с двоичной нумерацией:

$$f_0 = f(0,0,0), \quad f_1 = f(0,0,1), \quad f_2 = f(0,1,0), \quad f_3 = f(0,1,1),$$
$$f_4 = f(1,0,0), \quad f_5 = f(1,0,1), \quad f_6 = f(1,1,0), \quad f_7 = f(1,1,1).$$

Аналогично занумеруем строку коэффициентов многочлена Жегалкина:

$$c_0 = c_{000}, \quad c_1 = c_{001}, \quad c_2 = c_{010}, \quad c_3 = c_{011},$$
$$c_4 = c_{100}, \quad c_5 = c_{101}, \quad c_6 = c_{110}, \quad c_7 = c_{111}.$$

Сам многочлен $P(x_1, x_2, x_3)$ поэтому можно записать в виде

$$c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus c_3x_1x_2 \oplus c_4x_3 \oplus c_5x_1x_3 \oplus c_6x_2x_3 \oplus c_7x_1x_2x_3.$$

Разложим его по переменной x_3:

$$P(x_1, x_2, x_3) = P(x_1, x_2, 0) + x_3P(x_1, x_2, 1) = P_0(x_1, x_2) + x_3P_1(x_1, x_2).$$

Очевидно, что

$$P_0(x_1, x_2) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus c_3x_1x_2,$$
$$P_1(x_1, x_2) = c_4 \oplus c_5x_1 \oplus c_6x_2 \oplus c_7x_1x_2.$$

Так как

$$P_0(x_1, x_2) = P(x_1, x_2, 0) = f(x_1, x_2, 0),$$
$$P_0(x_1, x_2) \oplus P_1(x_1, x_2) = P(x_1, x_2, 1) = f(x_1, x_2, 1),$$

получаем

$$P_1(x_1, x_2) = f(x_1, x_2, 0) \oplus f(x_1, x_2, 1) = f_0(x_1, x_2) \oplus f_1(x_1, x_2),$$

где $f(x_1, x_2, i)$ для краткости обозначено $f_i = f_i(x_1, x_2)$. Обозначим через T преобразование, переводящее функцию f в реализующий ее многочлен Жегалкина P, а через t — соответствующее преобразование значений функции в его коэффициенты. Тогда, очевидно,

$$P_0 = T(f_0), P_1 = T(f_0 \oplus f_1),$$
$$(c_0, c_1, c_2, c_3) = t(f_0, f_1, f_2, f_3),$$
$$(c_4, c_5, c_6, c_7) = t(f_0 \oplus f_4, f_1 \oplus f_5, f_2 \oplus f_6, f_3 \oplus f_7).$$

Преобразование t состоит из 8 функций t_i от переменных f_i, $i = 0, \ldots, 7$, и каждая из них представляется многочленом Жегалкина первой

степени с нулевым свободным членом. Такие многочлены далее называются линейными (и само преобразование t тоже называется поэтому линейным). Любой из них можно записать в виде

$$t_i = c_{i0} \,\&\, f_0 \oplus \ldots \oplus c_{i7} \,\&\, f_7,$$

где коэффициенты c_{ij} — нули или единицы. Докажем это. Очевидно, что сумма по модулю два любых линейных функций будет линейной функцией, и для любой линейной функции $f(x_1, \ldots, x_n)$ справедливо тождество

$$f(x_1 \oplus y_i, \ldots, x_n \oplus y_n) = f(x_1, \ldots, x_n) \oplus f(y_1, \ldots, y_n).$$

Из этого тождества и линейности функций t_0, t_1, t_2, t_3 (см. задачу 25) с учетом равенства

$$(c_4, c_5, c_6, c_7) = t(f_0 \oplus f_4, f_1 \oplus f_5, f_2 \oplus f_6, f_3 \oplus f_7)$$

получаем, что и t_4, t_5, t_6, t_7 — линейные функции от переменных f_i, $i = 0, \ldots, 7$, причем коэффициенты у них при парах переменных t_i, t_{i+4}, $i = 0, 1, 2, 3$, одинаковые и совпадают с коэффициентами при переменных f_i, $i = 0, 1, 2, 3$, у функций t_0, t_1, t_2, t_3 соответственно. Отсюда можно сделать два вывода.

Во-первых, таблица коэффициентов c_{ij}, $i, j = 0, \ldots, 7$, линейных функций t_i, $i = 0, \ldots, 7$, — это квадрат размера 8×8, который разбивается средними линиями на 4 квадрата размера 4×4, один из которых нулевой (состоит из одних нулей), а три остальных — одинаковые и совпадают с матрицей C_2:

$$C_3 = \begin{pmatrix} C_2 & 0 \\ C_2 & C_2 \end{pmatrix} = \left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right).$$

Аналогичным образом можно получить таблицу C_4 размера 16×16 и вообще таблицу C_n размера $2^n \times 2^n$ коэффициентов линейного преобразования $t(f_0, \ldots, f_{2^n-1})$:

$$C_n = \begin{pmatrix} C_{n-1} & 0 \\ C_{n-1} & C_{n-1} \end{pmatrix}.$$

Непосредственно видно, что таблицы C_1, C_2, C_3, C_4 совпадают с уже знакомыми нам «таблицами Паскаля по модулю два», которые ино-

гда называют также матрицами Серпинского. По индукции можно доказать, что так будет и для любой таблицы C_n.

Во-вторых, можно заметить, что если обозначить через $L(n)$ число операций сложения по модулю два, используемых в указанном выше алгоритме для вычисления по значениям булевой функции n переменных коэффициентов ее многочлена Жегалкина, то $L(3) = 2L(2) + 4$, так как для вычисления

$$(c_0, c_1, c_2, c_3) = t(f_0, f_1, f_2, f_3)$$

нужно $L(2)$ операций, потом 4 операции для вычисления

$$f_0 \oplus f_4, \quad f_1 \oplus f_5, \quad f_2 \oplus f_6, \quad f_3 \oplus f_7$$

и еще $L(2)$ операции для вычисления

$$(c_4, c_5, c_6, c_7) = t(f_0 \oplus f_4,\ f_1 \oplus f_5,\ f_2 \oplus f_6,\ f_3 \oplus f_7).$$

Аналогично получается равенство $L(n) = 2L(n-1) + 2^{n-1}$. Очевидно, $L(1) = 1$.

Задача 26. Докажите по индукции, что $L(n) = 2^{n-1}n$.

Таким образом, для вычисления коэффициентов многочлена Жегалкина для данной булевой n-местной функции достаточно $2^{n-1}n$ битовых операций сложения по модулю два. Так как для этого вычисления надо использовать все 2^n значений данной функции, ясно, что указанный алгоритм если не оптимальный, то не очень далек от оптимального. Является ли он оптимальным, до сих пор неизвестно.

Задача 27. Докажите, что преобразование, обратное к преобразованию $c_i = t_i(f_0, \ldots, f_{2^n-1})$, то есть преобразование, вычисляющее по коэффициентам $c_0, \ldots, c_{2^n-1}$ многочлена Жегалкина от n переменных значения $f_0, \ldots, f_{2^n-1}$ реализуемой им функции, совпадает с исходным преобразованием.

И наконец, укажем одно применение рассмотренного алгоритма[12]. Его можно использовать для того, чтобы быстро умножать многочлены Жегалкина. Действительно, пусть даны два многочлена P_1, P_2 от n переменных. Нужно найти их произведение P_1P_2 и преобразовать его тоже в многочлен Жегалкина. Для этого можно перемножить каждую пару одночленов из P_1 и P_2, в полученном одночлене устра-

[12] Об этом автор узнал от профессора кафедры дискретной математики мехмата МГУ А. В. Чашкина.

нить квадраты, так как $x^2 = x$, а потом привести подобные члены, уничтожив одинаковые пары многочленов. Таким образом, получим многочлен, в котором не будет высоких степеней переменных и все одночлены будут различны, т. е. многочлен Жегалкина. Но число операций в подобном алгоритме будет не меньше 2^{2n}. Оказывается, есть другой простой алгоритм, выполняющий указанное умножение за $3 \cdot 2^{n-1}n + 2^n$ операций $\oplus$ и &. В нем нужно сначала со сложностью $2^n n$ по коэффициентам многочленов P_1, P_2 вычислить таблицу значений реализуемых ими булевых функций, потом, выполнив 2^n операций &, вычислить таблицу значений функции $P_1 \,\&\, P_2$ и с помощью $2^{n-1}n$ операций $\oplus$ восстановить по ним коэффициенты многочлена Жегалкина, являющегося произведением P_1 и P_2.

Заключительные разделы содержат некоторые дополнения к основной теме книжки.

7. Кое-что о записи чисел в двоичной системе

Для любого n обозначим через $\lambda(n)$ уменьшенную на единицу длину двоичной записи числа n, а $\nu(n)$ — ее сумму цифр (другими словами, число единиц в ней).

Очевидно, что $\lambda(n) + \nu(n) - 1 \leqslant 2\lambda(n)$. Те, кто знают логарифмы, сообразят, что $\lambda(n) = \lfloor \log_2 n \rfloor$, где знак $\lfloor x \rfloor$ означает целую часть числа x. Но можно вычислить обе введенные функции, даже не упоминая о двоичной записи. Для этого надо воспользоваться следующими правилами:

$$\nu(1) = 1, \quad \nu(2n) = \nu(n), \quad \nu(2n+1) = \nu(n) + 1,$$
$$\lambda(1) = 0, \quad \lambda(2n) = \lambda(2n+1) = \lambda(n) + 1.$$

Однако для доказательства справедливости этих правил полезно, конечно, воспользоваться двоичной системой, после чего они становятся почти очевидными. Заметим, что последнее правило позволяет рекуррентно вычислять $\lambda(n)$ и тем самым дает определение $\lambda(n)$, независимое от понятия двоичного логарифма (и, значит, дает представление о двоичном логарифме для тех, кто не знает логарифмов).

Задача 28. Докажите неравенство $\nu(n+1) \leqslant \nu(n) + 1$.

Указание. Оно, очевидно, превращается в равенство, если n четно, так как тогда его двоичная запись заканчивается нулем. Если же эта двоичная запись заканчивается k единицами, перед которыми

стоит нуль, то двоичная запись числа $n+1$ заканчивается k нулями, перед которыми стоит единица (а старшие биты остаются без изменения, если они есть). Для того, чтобы в этом убедиться, достаточно выполнить прибавление 1 к n в двоичной системе. В обоих рассмотренных случаях $\nu(n+1) \leqslant \nu(n)+1$.

Задача 29. Докажите неравенство $\lambda(n+1)+\nu(n+1)\leqslant\lambda(n)+\nu(n)+1$.

Указание. Действительно, если $2^{k-1} < n+1 < 2^k$, то $\lambda(n+1) = k-1 = \lambda(n)$, и из неравенства $\nu(n+1) \leqslant \nu(n)+1$ следует нужная оценка. Если же $n+1=2^k$, то $\lambda(n+1)=k=\lambda(n)+1$, $\nu(n+1)=1$, $\nu(n)=k$, откуда следует, что $\lambda(n+1)+\nu(n+1)=k+1\leqslant 2k=\lambda(n)+\nu(n)+1$.

Задача 30. Докажите равенство $\lambda(2n)+\nu(2n)=\lambda(n)+\nu(n)+1$.

Указание. Оно следует из равенств $\nu(2n)=\nu(n)$, $\lambda(2n)=\lambda(n)+1$.

8. Что такое булевы схемы

Для выполнения логических операций в электронике было создано множество устройств. Современные устройства, называемые логическими ячейками или элементами, имеют микроскопические размеры и представляют из себя особые участки кремниевого кристалла. Физика и химия протекающих в них процессов весьма сложна и не будет обсуждаться в этой книжке. Также сложна и технология их производства. Но логика работы этих элементов довольно проста: каждый из них реализует определенную логическую (или, как говорят, булеву) операцию. Элемент, реализующий (или вычисляющий) конъюнкцию, называется *конъюнктором*, на схемах иногда обозначается символом AND (рис. 4). Элемент дизъюнкции (*дизъюнктор*) на схемах иногда обозначается OR (рис. 5). Элемент, реализующий сумму по модулю два, на схемах обозначается XOR (сокращение от английского EXCLUSIVE OR — «исключающее или»; рис. 6). Для выполнения сложения однобитных чисел делают обычно даже специальный логический элемент с двумя входами x, y и двумя выходами w, v, как бы составленный из

Рис. 4. Элемент конъюнкции

Рис. 5. Элемент дизъюнкции

элемента умножения (конъюнкции) и элемента сложения по модулю два. Этот элемент часто называют *полусумматором* (рис. 7).

Рис. 6. Элемент суммы по модулю два («исключающее или»)

Мы не будем давать строгого определения понятия логической (или булевой) схемы, а всего лишь рассмотрим в качестве примера два варианта схемной реализации полусумматора — использующий элемент XOR и обходящийся без него. В первом варианте схема состоит из двух элементов, а во втором — из четырех. Число элементов в схеме далее будем называть ее *сложностью*. Сложность данной схемы S обычно обозначается $L(S)$. Сложностью схемы в значительной степени определяются ее физические размеры, т. е. площадь, занимаемая схемой на кремниевом кристалле. Как правило, чем больше сложность, тем больше площадь.

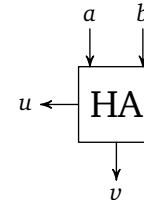


Рис. 7. Полусумматор

Другой важной характеристикой схемы является ее глубина. *Глубиной* схемы называется максимальное число ее элементов, образующих цепь, соединяющую какой-либо вход схемы с одним из ее выходов. Например, у схемы на рис. 8 глубина равна единице, а у схемы на рис. 13 — трем. Глубина схемы S обычно обозначается $D(S)$.

Глубина схемы в значительной степени определяет ее задержку. *Задержкой* схемы называется время, прошедшее от момента появления сигнала на входах схемы (как правило, значения входов схемы стабилизируются в разные момент времени, и тогда отсчет начинается с последнего из них) до момента появления сигнала на ее выходе. Как правило, чем больше глубина, тем больше задержка.

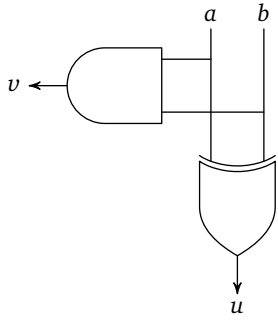


Рис. 8. Реализация полусумматора с помощью элемента XOR

Пример схемы из элементов AND и XOR, реализующей булев оператор HA (от английского Half Adder), приведен на рис. 8. Схему из элементов AND, OR, NOT (конъюнкций, дизъюнкций и отрицаний) для полусумматора можно построить, используя следую-

щую формулу: $x \oplus y = (\neg x \,\&\, y) \vee (x \,\&\, \neg y)$. Предлагается читателю нарисовать схему самостоятельно. В ней будет шесть элементов.

Задача 31. Проверьте справедливость тождества (проверить его гораздо проще, чем до него догадаться!)

$$x \oplus y = \neg(x \,\&\, y) \,\&\, (x \vee y).$$

Задача 32. Используя это тождество, постройте схему для полусумматора из четырех элементов.

9. Схемная реализация сумматора однобитных чисел

Сложность доказанной в предыдущих разделах формулы

$$f_3(x_1, \ldots, x_{15}) = \sum_{1 \leqslant i_1 < \ldots < i_8 \leqslant 15} x_{i_1} \ldots x_{i_8} = \\ = x_1 \ldots x_8 \oplus x_1 \ldots x_7 x_9 \oplus \ldots \oplus x_8 \ldots x_{15},$$

вычисляющей старший разряд суммы 15 битов, равна

$$8 \cdot \binom{15}{7} - 1 = 51479.$$

Задача 33. Проверьте это.

Указание. В многочлене Жегалкина $f_3(x_1, \ldots, x_{15})$ содержится ровно $\binom{15}{8} = \binom{15}{7}$ слагаемых. Поэтому общее число символов переменных в этой формуле равно $8 \cdot \binom{15}{7}$. Число элементов AND и XOR (символов операций $\&, \oplus$) в подобных формулах всегда на единицу меньше, чем число символов переменных.

Хотя многочлены f_0, f_1, f_2 для остальных разрядов суммы 15 битов существенно проще, вычисленная сложность очень велика и сравнима с размерами таблицы для всех четырех булевых функций, реализуемых этими многочленами. В общем случае суммы n битов сложность многочлена Жегалкина для старшего разряда этой суммы будет по порядку равна $2^n\sqrt{n}$. Это очень много. Кажется, что использование формул алгебры логики все-таки не дает существенного выигрыша в сравнении с простым применением таблиц. Но можно, оказывается, построить формулу, вычисляющую этот многочлен и содержащую не более $Cn^{3{,}03}$ символов переменных, где C — некоторая (довольно большая) константа. Эту рекордную формулу недавно придумал молодой московский математик Игорь Сергеев, научный

сотрудник кафедры дискретной математики мехмата МГУ. Привести ее построение здесь мы не можем, так как оно весьма сложно.

Но если для реализации упомянутых многочленов использовать не формулы, а булевы схемы (в которых, в отличие от формул, выходы элементов можно использовать многократно для присоединения их к входам других элементов), то сложность реализации всей схемы, вычисляющей сумму n битов, оказывается еще меньше. Такие схемы были давно известны, и построить их (в отличие от формул) несложно. Далее будет показано, как это сделать. Для краткости обозначаем эти схемы символом FA_n (от английского Full Adder).

Для примера построим схему FA_9. Представим ее в виде нескольких рисунков. На них изображены модули (подсхемы) этой схемы и на последнем — сама схема. Проверить, что эта схема работает правильно, читателю предлагается самостоятельно. Вам в этом помогут подписи к рис. 9–12 (для разных схем используются обозначения с разными верхними индексами).

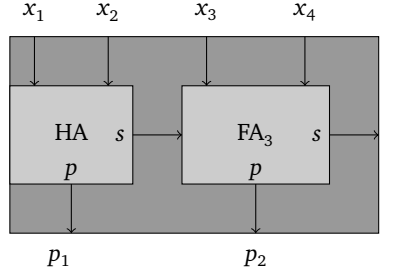


Рис. 9. Модуль FA_4^1. Входы и выходы схемы связаны соотношением $x_1 + x_2 + x_3 + x_4 = 2(p_1 + p_2) + s$

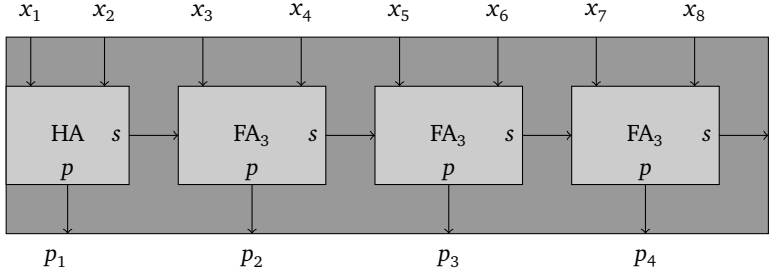


Рис. 10. Модуль FA_8^1. Входы и выходы схемы связаны соотношением $x_1 + \ldots + x_8 = 2(p_1 + \ldots + p_4) + s$

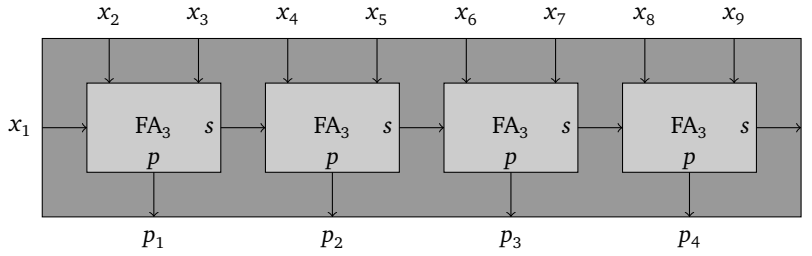


Рис. 11. Модуль FA_9^1. Входы и выходы схемы связаны соотношением $x_1 + \ldots + x_9 = 2(p_1 + \ldots + p_4) + s$

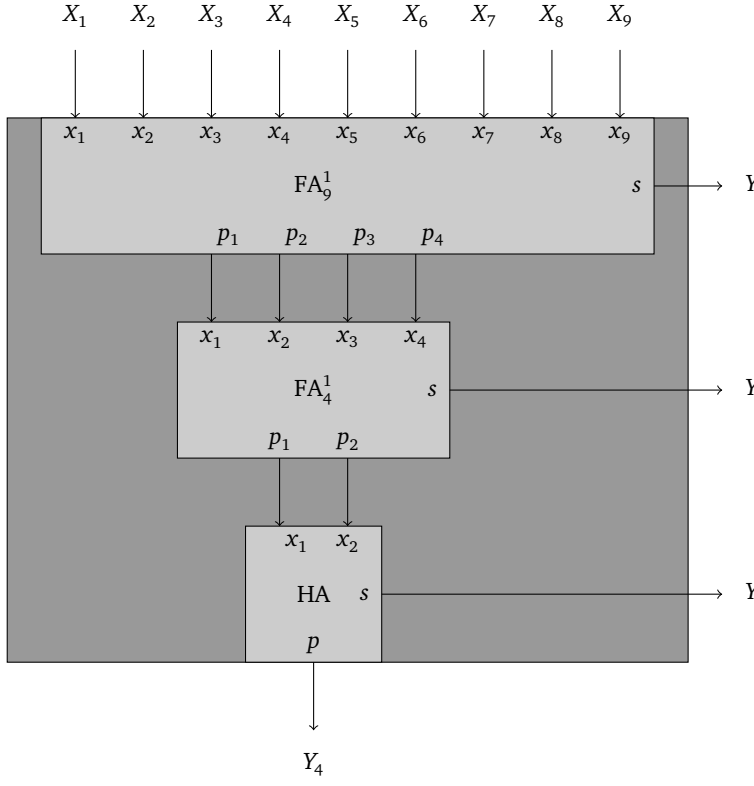


Рис. 12. Модуль FA_9. Входы и выходы схемы связаны соотношением $X_1 + \ldots + X_9 = 2^3 Y_4 + 2^2 Y_3 + 2Y_2 + Y_1$, то есть схема является счетчиком числа единиц в наборе $(X_1, \ldots, X_9)$

Задача 34. Схемы для FA_2 и FA_3 нарисуйте самостоятельно (одну из них можно увидеть на рис. 13).

Аналогично можно построить схемы и для FA_{15} (сумматора 15 битов) и вообще для сумматора n битов.

Задача 35. Докажите, что сложность схемы для сумматора n битов менее $5n$.

Недавно молодые петербургские математики А. С. Куликов и Е. А. Деменков (Санкт-Петербургское отделение МИРАН) построили схему для сумматора n битов, сложность которой асимптотически равна $4{,}5n$, а И. С. Сергеев построил еще лучшую схему, которая при почти такой же сложности имеет рекордно малую глубину, асимптотически равную $3{,}34\log_2 n$. Используя эту схему, он также построил схему для умножения n-битных чисел, сложность которой асимптотически равна $5{,}5n^2$, а глубина $4{,}34\log_2 n$. Эта глубина является в настоящий момент рекордной, но схемы для умножения с существенно большей глубиной могут иметь меньшую сложность (см. об этом, например, [1]).

10. Сумматор — схема для сложения двух двоичных чисел

Два n-разрядных двоичных числа $x = (x_n \dots x_1)_2$ и $y = (y_n \dots y_1)_2$ складываются школьным методом в столбик следующим образом:

$$\begin{array}{rr} & q_{n+1}q_n \dots q_1 \\ + & \phantom{q_{n+1}}x_n \dots x_1 \\ & \phantom{q_{n+1}}y_n \dots y_1 \\ \hline & z_{n+1}z_n \dots z_1. \end{array}$$

Числа $q_1, \dots, q_{n+1}$ — результаты переносов. Первый настоящий перенос q_2 возникает только тогда, когда $x_1 = y_1 = 1$. Каждый следующий перенос q_{i+1} возникает только тогда, когда суммируемые разряды x_i, y_i и перенос q_i из предыдущего разряда в сумме дают число не менее двух, т. е. среди этих трех чисел не меньше двух единиц. Только в этом случае $q_{i+1} = 1$, в противном случае $q_{i+1} = 0$. Булева функция от трех переменных x, y, z, которая обращается в 1, только когда $x + y + z \geqslant 2$, является функцией голосования комитета из трех человек. Ее называют также мажоритарной функцией, или медианой. Ее можно выразить через знакомые нам булевы функции двух переменных формулой $m(x, y, z) = xy \vee xz \vee yz$ или $xy \oplus xz \oplus yz$. Обе формулы

правильно вычисляют (или, как еще говорят, реализуют) медиану $m(x, y, z)$, в чем легко убедится непосредственной проверкой. Если $x = y = z = 1$, то обе формулы дают в результате $1 = m(1, 1, 1)$ (действительно, $1 \vee 1 \vee 1 = 1 = 1 \oplus 1 \oplus 1$). Если же, например, $x = y = 1$, $z = 0$, то в обоих формулах ровно одно слагаемое из трех равно единице, значит, и в этом случае обе формулы вычисляют $1 = m(1, 1, 0)$. Два оставшихся случая рассматриваются точно так же, причем в силу симметричности обеих формул и функции m ничего нового при их рассмотрении не возникнет. Пользуясь полученными формулами, легко заметить, что сложение трех однобитных чисел x, y, z можно выполнить, пользуясь формулами:

$$x + y + z = 2u + v, \quad v = x \oplus y \oplus z, \quad u = m(x, y, z) = xy \oplus yz \oplus xz.$$

Из сказанного ясно, что сложение двух n-разрядных чисел выполняется по следующим формулам (в которых $i = 1, 2, \ldots, n$):

$$\begin{cases} q_1 = 0, \\ z_i = x_i \oplus y_i \oplus q_i, \\ q_{i+1} = x_i y_i \oplus (x_i \oplus y_i) q_i, \\ z_{n+1} = q_{n+1}. \end{cases}$$

Обозначим через B_i, где $i > 1$, изображенную на рис. 13 схему сложности пять и глубины три.

Нужная нам схема A_n — сумматор двух n-разрядных двоичных чисел — получается путем последовательного соединения блоков B_i, $i = 1, \ldots, n$, как показано на рис. 14.

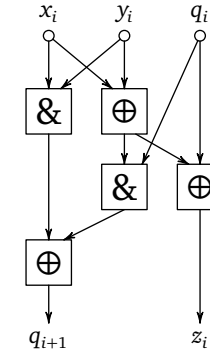


Рис. 13. Модуль вычисления очередного разряда и переноса

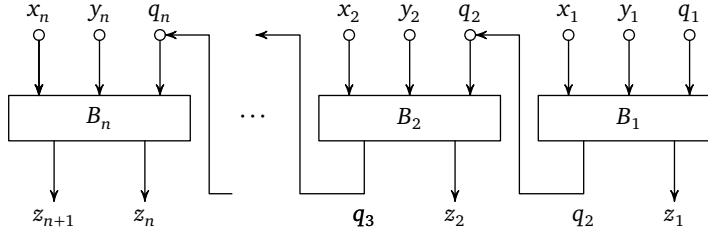


Рис. 14. Схема n-разрядного двоичного сумматора

В этой схеме блок B_1 выполняет преобразование $z_1{=}x_1{+}y_1 \pmod 2$, $q_2 = x_1 y_1$ сложности два и глубины один. Таким образом, сложность построенного сумматора $L(A_n) = 5n - 3$, а глубина $D(A_n) = 3n - 2$.

Построенная схема является оптимальной (т. е. имеет минимальную сложность), как показал около сорока лет назад Н. П. Редькин (в настоящее время профессор кафедры дискретной математики мехмата МГУ). Но глубина этой схемы минимальной не является. Задача построения сумматора минимальной глубины оказалась крайне трудной. Схему асимптотически минимальной глубины построил в конце 1960-х годов В. М. Храпченко, ведущий научный сотрудник сектора теоретической кибернетики ИПМ РАН. Недавно доцент кафедры дискретной математики мехмата МГУ М. И. Гринчук построил чуть лучшую схему, глубину которой можно оценить сверху простой формулой

$$\log_2 n + \log_2 \log_2 n + 6.$$

Наилучшая нижняя оценка глубины такой схемы (если она построена только из элементов $\&, \vee, \neg$), недавно полученная В. М. Храпченко, имеет вид

$$\log_2 n + 0{,}15 \log_2(\log_2 \log_2 n).$$

11. Подсчет числа единиц в булевой строке компьютером

Встречаются ситуации, когда компьютеру приходится выполнять операции не с числами, а с битами, составляющими эти числа (или машинные слова). Для этого используются команды языка ассемблера. Но и язык C (и C++) имеет средства для непосредственной работы с битами. Для быстрого выполнения различных манипуляций с битами в программистском фольклоре и в литературе известно множество различных эффективных программ. Замечательная коллекция таких трюков содержится в [9] (английский оригинал называется «Hacker's Delight»).

Рассмотрим задачу подсчета числа единичных битов в машинном слове (т. е. количества единичных разрядов в данном 32-битном числе x). По существу, это та же задача, с которой мы имели дело в предыдущих разделах, но решать ее здесь мы будем не с помощью булевых формул и схем, а с помощью компьютерных команд и программ. Известно много элегантных и эффективных решений этой задачи. Приведем одно из лучших.

Заметим, что рассматриваемая задача очень близка к задаче построения логической схемы, выполняющей суммирование n однобитных чисел в двоичной системе счисления. Известно, что такую схему можно построить следующим образом. Разобьем эти числа на пары и сложим каждую пару. Получим $n/2$ двухбитных чисел. Эти числа опять разобьем на пары и сложим каждую пару. Получим $n/4$ трехбитных чисел. Разобьем эти числа на пары и сложим каждую пару. Получим $n/8$ четырехбитных чисел. Далее получаем $n/16$ пятибитных чисел и т. д. В результате получится одно $(\log_2 n + 1)$-битное число (предполагаем, что n есть степень двойки), которое и равно сумме всех n чисел, а другими словами — количеству единиц среди них.

Указанную схему при $n = 32$ можно промоделировать следующей компьютерной программой. Сначала вычисляем $x \,\&\, (010101\ldots0101)_2$ и получаем число $(0x_{31}0x_{29}\ldots0x_1)_2$. Аналогично получаем число $(0x_{32}0x_{30}\ldots0x_2)_2$ в результате операции

$$x \leftarrow (x \gg 1) \,\&\, (010101\ldots0101)_2,$$

где $(x \gg 1)$ означает сдвиг битов числа x на один разряд вправо. Потом складываем полученные числа. Все вместе это выполняется следующим образом:

$$x \leftarrow x \,\&\, (010101\ldots0101)_2 + (x \gg 1) \,\&\, (010101\ldots0101)_2.$$

В результате получится число $(y_{32}\ldots y_1)_2$, для которого

$$(y_{2i}y_{2i-1})_2 = x_{2i} + x_{2i-1}, \quad i = 1, \ldots, 16.$$

Значит, выполнив 4 команды, мы параллельно сложили 16 пар чисел. Далее выполняем команды:

$$x \leftarrow (x \,\&\, (00110011\ldots0011)_2) + ((x \gg 2) \,\&\, (00110011\ldots0011)_2).$$

Результат состоит из четверок битов, которые являются двоичными записями восьми сумм

$$x_{4i} + x_{4i-1} + x_{4i-2} + x_{4i-3}, \quad i = 1, \ldots, 8.$$

Заметим, что первый бит в каждой четверке нулевой, так как сумма четырех однобитных чисел трехбитна. Далее выполняем аналогичным образом команды:

$$x \leftarrow (x \,\&\, (00001111\ldots00001111)_2) + \\ + ((x \gg 4) \,\&\, (00001111\ldots00001111)_2).$$

Результат состоит из восьмерок битов, которые являются двоичными записями четырех сумм

$$x_{8i} + x_{8i-1} + \ldots + x_{8i-7}, \quad i = 1, \ldots, 4.$$

Заметим, что левая половина битов в каждой восьмерке нулевая, так как сумма восьми однобитных чисел четырехбитна. Далее выполняем аналогичным образом команды:

$$x \leftarrow (x \,\&\, (00000000111111110000000011111111)_2) + \\ + ((x \gg 8) \,\&\, (00000000111111110000000011111111)_2).$$

В результате получится число, состоящее из двух блоков по 16 бит, которые являются двоичными записями двух сумм

$$x_{16i} + x_{16i-1} + \ldots + x_{16i-15}, \quad i = 1, 2.$$

Заметим, что левая половина битов в каждом блоке нулевая, так как сумма 16 однобитных чисел пятибитна. Далее выполняем аналогичным образом команды:

$$x \leftarrow (x \,\&\, (00000000000000001111111111111111)_2) + \\ + ((x \gg 16) \,\&\, (00000000000000001111111111111111)_2)$$

и получаем окончательно нужную нам сумму, в которой левая половина битов нулевая, а ненулевыми могут быть только правые 6 битов. Заметим, что каждую строку, кроме первой и второй, можно упростить, устраняя первую конъюнкцию и перегруппировывая скобки. Например, третью скобку можно переписать в виде

$$x \leftarrow (x + (x \gg 4)) \,\&\, (00001111 \ldots 00001111)_2.$$

При этом результат работы каждой строки не изменится, так как каждый из четырехбитных блоков, на которые разбивается число x, полученное в результате работы первых двух строк, имеет нуль в самом левом бите, поэтому при сложении этих четырехбитных блоков не происходит переноса в другие блоки. Первую строку можно упростить, заменив на следующую:

$$x \leftarrow x - ((x \gg 1) \,\&\, (0101 \ldots 01)_2).$$

Результат при этом не изменится, потому что при вычитании в каждой паре соседних битов будет производиться операция

$$2^{2i}(2x_{2i+1} + x_{2i}) - x_{2i+1}2^{2i} = 2^{2i}(x_{2i+1} + x_{2i}).$$

Далее, эту программу можно несколько упростить, заменив последнюю строку на следующую:

$$x \leftarrow x + (x \gg 16).$$

Полученный результат будет совпадать с правильным только в 16 правых битах, но нужная нам информация содержится именно в них, точнее, даже в 6 самых правых. Поэтому для правильной работы программы достаточно добавить строку

$$\text{return } x \,\&\, (00\ldots0111111)_2.$$

Аналогичным образом можно заменить и предпоследнюю строку в исходной программе на

$$x \leftarrow x + (x \gg 8).$$

Результат тогда будет совпадать с правильным в самой правой восьмерке битов, в следующей восьмерке он, возможно, будет неправильным (ненулевым), а в следующей — опять правильным, причем сумма этих правильных результатов (чисел, двоичные записи которых задают указанные восьмерки) будет окончательным результатом программы. Поэтому если выполнять уже измененные строки

$$x \leftarrow x + (x \gg 16), \quad \text{return } x \,\&\, (00\ldots0111111)_2,$$

окончательный результат все равно будет верным. Известно также много других эффективных программ для решения той же задачи. Некоторые из них основаны на следующих формулах, которые предлагаются в виде задач.

Задача 36. Обозначим сумму битов n-битного числа x через $\|x\|$. Докажите, что

$$\|x\| = x - \left\lfloor \frac{x}{2} \right\rfloor - \left\lfloor \frac{x}{4} \right\rfloor - \ldots - \left\lfloor \frac{x}{2^{n-1}} \right\rfloor.$$

Задача 37. Обозначим $(x \ll i)^{\text{rot}}$ циклический битовый сдвиг на i позиций. Докажите, что

$$\|x\| = -\sum_{i=0}^{n-1} (x \ll i)^{\text{rot}} \mod 2^n.$$

Указание. Каждый бит при сдвигах пробегает все возможные позиции и сумма этих чисел по модулю 2^n равна $x(11\ldots1)_2 \bmod 2^n = -x$.

В качестве применения быстрого алгоритма вычисления $\|x\|$ можно быстро вычислить знакопеременную сумму битов

$$s(x) = x_1 - x_2 + x_3 - \ldots + x_{31} - x_{32}.$$

Очевидно,

$$s(x) = \|x \,\&\, (0101\ldots01)\| - \|x \,\&\, (1010\ldots10)\|, \quad -16 \leqslant s(x) \leqslant 16.$$

Остаток от деления неотрицательного числа x на 3 можно найти, не выполняя деления, вычислив $s(x)$ и заметив, что $x \bmod 3 = s(x) \bmod 3$. Далее быстрее всего воспользоваться предвычисленной таблицей остатков по модулю три для чисел от -16 до 16. В случае знаковых чисел в определении $s(x)$ нужно заменить $-x_{32}$ на $+x_{32}$. Тогда в формулу для вычисления $s(x)$ надо добавить слагаемое $((x \gg 31) \ll 1)$, если компьютер выполняет сдвиг знаковых чисел вправо с заполнением слева нулями, а если такой команды нет, но есть команда правого знакового сдвига, при котором слева все биты заполняются знаковым битом, тогда надо вычесть это слагаемое.

Для вычисления остатка от деления x на 3 можно также положить

$$s(x) = x_1 + 2x_2 + x_3 + 2x_4 + \ldots + x_{31} + 2x_{32}$$

(для положительных чисел). Эта формула вычисляется чуть быстрее:

$$s(x) = \|x\| + \|x \,\&\, (10\ldots10)_2\|.$$

Но размеры используемой далее таблицы немного возрастут.

Подобный же прием можно использовать и при вычислении остатка от деления на 7, но в нем уже придется три раза применять функцию $\|x\|$. Возможно, в этом случае более быстрой окажется программа, вычисляющая

$$s(x) = x_1 + 2x_2 + 4x_3 + x_4 + 2x_5 + 4x_6 + \ldots$$

по формуле

$$s(x) = (x \gg 30) \,\&\, (0\ldots0111)_2 + \ldots \\ \ldots + (x \gg 3) \,\&\, (0\ldots0111)_2 + x \,\&\, (0\ldots0111)_2.$$

Остаток от деления на 15 можно вычислять с помощью функции

$$s(x) = x_1 + 2x_2 + 4x_3 + 8x_4 + x_5 + 2x_6 + 4x_7 + 8x_8 + \ldots,$$

задаваемой формулой

$$s(x) = (x \gg 28) \,\&\, (0\ldots01111)_2 + \ldots \\ \ldots + (x \gg 4) \,\&\, (0\ldots01111)_2 + x \,\&\, (0\ldots01111)_2,$$

еще быстрее. Правда, размер используемой в конце таблицы возрастет до 120 чисел. Если заменить составляющие ее числа от нуля до 15

на их остатки по модулю три или пять, получим быстрые программы вычисления $x \bmod 3$ и $x \bmod 5$. Возможно, первая из них будет быстрее, чем указанная выше.

Литература

1. *Гашков С. Б.* Системы счисления и их применение. М.: МЦНМО, 2012.
2. *Гашков С. Б.* Занимательная компьютерная арифметика. Математика и искусство счета на компьютерах и без них. М.: Книжный дом «Либроком»/URSS, 2012.
3. *Гашков С. Б.* Занимательная компьютерная арифметика. Быстрые алгоритмы вычислений с числами и многочленами. М.: Книжный дом «Либроком»/URSS, 2012.
4. *Гашков С. Б.* Современная элементарная алгебра в задачах и упражнениях. М.: МЦНМО, 2006.
5. *Виленкин Н. Я.* Рассказы о множествах. М.: МЦНМО, 2013.
6. *Леман А. А.* Московские математические олимпиады. М.: Просвещение, 1965.
7. Квант № 4 за 2012 г. (Задачи LXXV Моск. матем. олимпиады, 11 класс, второй день).
8. *Гашков С. Б.*, *Чубариков В. Н.* Арифметика, алгоритмы, сложность вычислений. М.: Дрофа, 2005.
9. *Уоррен Г.* Алгоритмические трюки для программистов. 2-е изд, расширенное. М.: Вильямс, 2014.

Оглавление

Гашков Сергей Борисович

Сложение однобитных чисел

Треугольник Паскаля, салфетка Серпинского
и теорема Куммера

Подписано в печать 13.03.2014 г. Формат $60\times90\,{}^{1}\!/_{16}$. Бумага офсетная.
Печать офсетная. Печ. л. 2,5. Тираж 1000 экз. Заказ №

Издательство Московского центра
непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11. Тел. (499) 241–74–83.

Отпечатано в ППП «Типография „Наука“».
121099, Москва, Шубинский пер., 6.
