Занятие 16

Из задач Занятия 8 нам известна дизъюнктивная нормальная форма для функции MAJ (напомним, эта функция принимает значение 1 тогда и только тогда, когда больше половины ее аргументов равны 1):

$$\mathrm{MAJ}(x_1,\ldots,x_n)=\bigvee_{\substack{1\leqslant i_1<\ldots<i_k\leqslant n\\ \frac{n}{2}<k\leqslant n}}(x_{i_1}\wedge x_{i_2}\wedge\ldots\wedge x_{i_k}).$$

Эта формула может быть даже сокращена:

$$\mathrm{MAJ}(x_1,\ldots,x_n)=\bigvee_{1\leqslant i_1<\ldots<i_{\lfloor\frac{n}{2}\rfloor+1}\leqslant n}(x_{i_1}\wedge x_{i_2}\wedge\ldots\wedge x_{i_{\lfloor\frac{n}{2}\rfloor+1}}).$$

В самом деле, все члены второй дизъюнкции содержатся в первой. С другой стороны, если для какого-то $k>\frac{n}{2}$ истинен дизъюнкт $x_{i_1}\wedge x_{i_2}\wedge\ldots\wedge x_{i_k}$, то из него можно выделить (истинную) конъюнкцию $\lfloor\frac{n}{2}\rfloor+1$ переменных, поскольку $\lfloor\frac{n}{2}\rfloor+1$ есть наименьшее целое число, строго большее $\frac{n}{2}$.

Вторая из приведенных формул содержит $\binom{n}{\lfloor\frac{n}{2}\rfloor+1}-1$ дизъюнкцию. Для дальнейшего, полезно отметить, что эта величина как функция n растет быстрее любого многочлена.

Для целых неотрицательных n положим $C(n)=\binom{n}{\lfloor\frac{n}{2}\rfloor}$. Легко заметить, что $C(n)=\binom{n}{\lceil\frac{n}{2}\rceil}$, поскольку $\lfloor\frac{n}{2}\rfloor+\lceil\frac{n}{2}\rceil=n$. Вообще же, если $n=2k$, имеем $\lfloor\frac{n}{2}\rfloor=\lceil\frac{n}{2}\rceil=k$ и, если $n=2k+1$, $\lfloor\frac{n}{2}\rfloor=k$ и $\lceil\frac{n}{2}\rceil=k+1$.

Лемма 1. *Для всех целых неотрицательных n верно*

$$C(n+1)=C(n)\cdot\frac{n+1}{\lceil\frac{n+1}{2}\rceil}.$$

Доказательство. Расписывая биномиальные коэффициенты через факториалы, получаем

$$\frac{C(n+1)}{C(n)}=(n+1)\cdot\frac{\lfloor\frac{n}{2}\rfloor!\lceil\frac{n}{2}\rceil!}{\lfloor\frac{n+1}{2}\rfloor!\lceil\frac{n+1}{2}\rceil!}.$$

Разбирая отдельно случаи $n=2k$ и $n=2k+1$, видим, что в каждом из них второй множитель справа равен $\frac{1}{k+1}$, а с другой стороны, $k+1=\lceil\frac{n+1}{2}\rceil$. □

Лемма 2. *Для всех целых неотрицательных n верно*

$$C(n) \geqslant \frac{2^n}{n+1}.$$

Доказательство. Индукция по n. Случай $n = 0$ тривиален. Сделав предположение для n, имеем

$$C(n+1) = C(n) \cdot \frac{n+1}{\lceil \frac{n+1}{2} \rceil} \geqslant \frac{2^n}{\lceil \frac{n+1}{2} \rceil} \geqslant \frac{2^{n+1}}{n+2}.$$

Поясним последнее неравенство. Оно, очевидно, эквивалентно $\lceil \frac{n+1}{2} \rceil \leqslant \frac{n}{2} + 1$. А это последнее, если $n = 2k$, равносильно $k + 1 \leqslant k + 1$, или $k + 1 \leqslant k + \frac{3}{2}$, если $n = 2k + 1$. □

Несложные выкладки, подобные приведенным, показывают, что

$$C(n+1) = \binom{n}{\lfloor \frac{n}{2} \rfloor + 1} \cdot \frac{n+1}{\lceil \frac{n}{2} \rceil}.$$

Поэтому, если бы выполнялось $\binom{n}{\lfloor \frac{n}{2} \rfloor + 1} = O(n^K)$ для некоторого натурального K, было бы верно $C(n) = O(n^{K+1})$, что противоречит, как легко проверить, лемме 2. Итак, размер рассмотренной выше ДНФ для функции MAJ не является полиномиально ограниченным.

Задача 16.1. Постройте схему полиномиального размера для функции $\mathrm{MAJ}(x_1, \ldots, x_n)$.

Решение. Выше мы видели, что построение монотонной схемы (т. е. в базисе $\{\vee, \wedge\}$) полиномиального размера для функции MAJ не слишком очевидно. Однако, при наличии отрицания задача упрощается.

Одна из возможностей в том, чтобы сложить как двоичные числа наборы $0 \ldots 0x_i$ длины n для всех x_i, а затем сравнить результат с фиксированным для каждого n числом $\lfloor \frac{n}{2} \rfloor + 1$. Как мы знаем из Лекций, каждое сложение можно реализовать схемой размера $O(n)$ (константу 0 вычислим схемой размера $O(1)$: $0 = x_1 \wedge \neg x_1$). В задаче 16.2 будет показано, что сравнение наборов длины n как двоичных чисел реализуемо схемой размера $O(n)$. Итого, получится схема размера $O(n^2)$.

Мы, однако, пойдем другим путем. Именно, используем алгоритм «сортировки пузырьком» для входного набора $\vec{x}$. В отсортированном по неубыванию наборе $\vec{w}$ более половины единиц будет, очевидно, тогда и только тогда, когда $w_{\lceil \frac{n}{2} \rceil} = 1$ (напомним, индексы аргументов начинаются с 1). С другой стороны, $\mathrm{MAJ}(\vec{x}) = \mathrm{MAJ}(\vec{w})$, поскольку наша функция симметрическая.

Прежде всего, рассмотрим схему сравнения двух битовых аргументов — «битовый компаратор» — размера $O(1)$. Схема Cmp имеет два входа x и y и три выхода l, e и g, из которых первый вычисляет предикат[1] «$x < y$», второй вычисляет «$x = y$»,

[1] т. е. «свойство» входов, которое может быть истинно или ложно; поскольку наши входы биты, предикаты здесь суть булевы функции.

а третий — «$x > y$». Итак, полагаем

$$\begin{aligned} l &= \neg x \wedge y \\ g &= x \wedge \neg y \\ e &= \neg(l \vee g). \end{aligned}$$

При «сортировке пузырьком» мы сравниваем соседние биты и меняем их местами, если порядок не правилен. Это реализует схема Swp со входами x, y и выходами x', y', для которых выполняется $x' \leqslant y'$:

$$\begin{aligned} c &= (\mathrm{Cmp}(x, y))_g \\ x' &= (\neg c \wedge x) \vee (c \wedge y) \\ y' &= (\neg c \wedge y) \vee (c \wedge x). \end{aligned}$$

(Запись в первой строке означает, что мы используем выход g схемы Cmp). Отметим также, что схемы для *конкретных* булевых функций — а не их семейств, параметризованных n, — мы могли бы не указывать явно, сославшись на вычислимость каждой функции некоторой схемой (не зависящего от n размера $O(1)$). Однако, мы предпочли большую наглядность.

Теперь реализуем один «проход» «пузырьковой сортировки», после которого в конце массива будет стоять элемент наибольшего значения. Рассмотрим схему Pass с n входами $\vec{z}$ и n выходами $\vec{z}\,'$

$$\begin{aligned} (z'_1, s_2) &= \mathrm{Swp}(z_1, z_2) \\ (z'_2, s_3) &= \mathrm{Swp}(s_2, z_3) \\ (z'_3, s_4) &= \mathrm{Swp}(s_3, z_4) \\ &\dots \\ (z'_{n-2}, s_{n-1}) &= \mathrm{Swp}(s_{n-2}, z_{n-1}) \\ (z'_{n-1}, z'_n) &= \mathrm{Swp}(s_{n-1}, z_n). \end{aligned}$$

(Запись $(s, t) = \mathrm{Swp}(x, y)$ мы понимаем как $s = (\mathrm{Swp}(x, y))_{x'},\ t = (\mathrm{Swp}(x, y))_{y'}$, т. е. мы присваиваем набору «переменных» (s, t) значения выходов схемы Swp, взятых в оговоренном ранее порядке: сначала x', затем y'.) Как видно, размер данной схемы $O(n)$. После n проходов нашего алгоритма мы получим отсортированный массив:

$$\begin{aligned} \vec{z}^{\,1} &= \mathrm{Pass}(\vec{x}) \\ \vec{z}^{\,2} &= \mathrm{Pass}(\vec{z}^{\,1}) \\ &\dots \\ \vec{w} &= \mathrm{Pass}(\vec{z}^{\,n-1}) \\ \mathrm{MAJ}(\vec{x}) &= w_{\lceil \frac{n}{2} \rceil}. \end{aligned}$$

Итак, мы построили схему размера $O(n^2)$. □

Задача 16.2. Булева функция сравнения $L(x_1, \dots, x_n; y_1, \dots, y_n)$ равна 1 тогда и только тогда, когда $(\overline{x_n \dots x_1})_2 < (\overline{y_n \dots y_1})_2$. Постройте схему размера $O(n)$, которая вычисляет $L(x_1, \dots, x_n; y_1, \dots, y_n)$.

Решение. Итак, нам требуется сравнить n-разрядные двоичные числа, представленные наборами $\vec{x}$ и $\vec{y}$. Мы считаем, что бит x_n представляет значение старшего разряда, а x_1 — младшего. Сравнивая два числа одинаковой разрядности, мы сначала сравниваем их старшие разряды. Если один меньше другого, то такое же отношение будет и между числами. Если же значения старших разрядов равны, результат сравнения тот же, что у чисел $(\overline{x_{n-1}\dots x_1})_2$ и $(\overline{y_{n-1}\dots y_1})_2$.

Соответственно, строим схему L_n, вычисляющую искомую функцию для n-разрядных чисел, рекурсией по n. Полагаем

$$L_1(x_1, y_1) \;=\; (\mathrm{Cmp}(x_1, y_1))_l$$

и, для всех $n \in \mathbb{N}$,

$$\begin{aligned} l_{n+1} &= (\mathrm{Cmp}(x_{n+1}, y_{n+1}))_l \\ e_{n+1} &= (\mathrm{Cmp}(x_{n+1}, y_{n+1}))_e \\ L_{n+1}(x_1, \dots, x_{n+1}, y_1, \dots, y_{n+1}) &= l_{n+1} \vee (e_{n+1} \wedge L_n(x_1, \dots, x_n, y_1, \dots, y_n)). \end{aligned}$$

Индукцией по n покажем, что размер схемы L_n будет $O(n)$. Точнее, мы установим, что $\mathrm{size}(L_n) \leqslant 10n$ для всех $n \in \mathbb{N}$. При $n = 1$ используем, что $\mathrm{size}(\mathrm{Cmp}) \leqslant 10$. Далее,

$$\mathrm{size}(L_{n+1}) = 4 + \mathrm{size}(L_n) \leqslant 4 + 10n \leqslant 10(n+1).$$

(Напомним, по определению из Учебника, число входов учитывается в размере, а у L_{n+1} на два входа больше, чем у L_n.) □

Задача 16.3. Постройте схему размера $O(n)$ для вычитания n-битовых целых чисел.

Решение. Пусть n-разрядное целое число x представляется набором $\vec{x} = x_0 x_1 \dots x_n$, где x_1 есть значение младшего, а x_n — старшего разряда, причем x_0 представляет знак числа, так что $x_0 = 1$ означает «минус». Мы будем считать, что 0 может быть представлен как со знаком «плюс», так и «минус».

Прежде всего, построим схему вычитания *беззнаковых* n-разрядных целых чисел. Воспользуемся вычитанием «в столбик». Этот алгоритм сводится к последовательности побитовых «вычитаний», каждое из которых принимает на вход уменьшаемое x, вычитаемое y и заимствование из данного разряда b, а на выход выдает разность d и заимствование из следующего разряда b'. Входы и выходы, по определению этого алгоритма, связаны следующим образом:

x	y	b	d	b'
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Как видно, можно положить:

$$\begin{aligned} d &= x \oplus y \oplus b \\ b' &= (\neg x \wedge (y \vee b)) \vee (x \wedge y \wedge b) \end{aligned}$$

Указание соответствующей схемы Diff размера $O(1)$ не вызывает трудностей. Теперь определим схему Diff_n вычитания неотрицательного n-разрядного числа y из x. На выходе получится набор $\vec{z}$ длины $n+1$, где бит z_0 представляет знак разности. Отметим, что знак будет «плюс» (т. е. $z_0 = 0$) в случае нулевой разности.

$$\begin{aligned} 0 &= x_1 \wedge \neg x_1 \\ (z_1, b_1) &= \mathrm{Diff}(x_1, y_1, 0) \\ (z_2, b_2) &= \mathrm{Diff}(x_2, y_2, b_1) \\ &\dots \\ (z_{n-1}, b_{n-1}) &= \mathrm{Diff}(x_{n-1}, y_{n-1}, b_{n-2}) \\ (z_n, z_0) &= \mathrm{Diff}(x_n, y_n, b_{n-1}) \end{aligned}$$

Очевидно, построенная схема имеет размер $O(n)$. Теперь укажем, как вычитать произвольные целые числа. Основное наблюдение состоит в том, что такое вычитание иногда сводится к сложению беззнаковых целых (для которого схема Sum_n размера $O(n)$, выдающая $(n+1)$-битовое беззнаковое целое, была построена на Лекции). Точнее,

$\operatorname{sgn} x$	$\operatorname{sgn} y$	$x - y$
$-$	$-$	$-(\lvert x\rvert - \lvert y\rvert)$
$-$	$+$	$-(\lvert x\rvert + \lvert y\rvert)$
$+$	$-$	$\lvert x\rvert + \lvert y\rvert$
$+$	$+$	$\lvert x\rvert - \lvert y\rvert$

Вычисление модулей и умножение на -1 получаются обнулением или обращением знакового бита соответственно, что требует схемы размера $O(1)$. Однако, какое же из значений в правом столбце подать на выход?

В отличие от многих языков программирования, в схемах в явном отсутствуют условные переходы. Чтобы реализовать таковой, можно провести вычисления при каждом из возможных условий, а затем, «обнулить» результаты, не соответствующие условию, выполненному в действительности.

Для наборов $\vec{a}$ и $\vec{b}$ длины n обозначим $\vec{a}$ && $\vec{b} = (a_1 \wedge b_1, \dots, a_n \wedge b_n)$ и $\vec{x}$ || $\vec{y} = (a_1 \vee b_1, \dots, a_n \vee b_n)$. Для бита c будем писать $\vec{a}$ && c вместо $\vec{a}$ && $(c, \dots, c)$. Аналогично в случае ||. Легко видеть, что указанные функции реализуются схемами размера $O(n)$. «Обнуление» будет основано на тождествах $\vec{a}$ && $1 = \vec{a}$, $\vec{a}$ && $0 = \vec{0}$ и $\vec{a}$ || $\vec{0} = \vec{a}$.

Итак, итоговая схема имеет вид:

$$\begin{aligned}
0 &= x_1 \wedge \neg x_1 \\
(d_0, d_1, \ldots, d_n) &= \mathrm{Diff}_n(x_1, \ldots, x_n, y_1, \ldots, y_n) \\
d_{n+1} &= 0 \\
(s_1, \ldots, s_n, s_{n+1}) &= \mathrm{Sum}_n(x_1, \ldots, x_n, y_1, \ldots, y_n) \\
s_0 &= 0 \\
\vec{s}\,' &= (\neg s_0, s_1, \ldots, s_{n+1}) \\
\vec{d}\,' &= (\neg d_0, d_1, \ldots, d_{n+1}) \\
x - y &= (\vec{d}\ \&\&\ (\neg x_0 \wedge \neg y_0)) \mid\mid (\vec{s}\ \&\&\ (\neg x_0 \wedge y_0)) \mid\mid \\
&\quad (\vec{s}\,' \ \&\&\ (x_0 \wedge \neg y_0)) \mid\mid (\vec{d}\,' \ \&\&\ (x_0 \wedge y_0)).
\end{aligned}$$

Очевидно, размер этой схемы $O(n)$. □

Задача 16.4. Пусть $n = k + 2^k$. Указательная функция $f(x_1, \ldots, x_k, y_0, \ldots, y_{2^k-1})$ равна y_x, где x — число, двоичная запись которого $x_k \ldots x_1$ (x_k значение старшего разряда, а x_1 — младшего). Постройте схему полиномиального от n размера для указательной функции.

Решение. Обозначим функции данного нам семейства f_k в зависимости от k. Нетрудно заметить, что при $k > 1$ имеет место равенство

$$f_k(x_1, \ldots, x_k, y_0, \ldots, y_{2^k-1}) = \begin{cases} f_{k-1}(x_1, \ldots, x_{k-1}, y_0, \ldots, y_{2^{k-1}-1}), & \text{если } x_k = 0; \\ f_{k-1}(x_1, \ldots, x_{k-1}, y_{2^{k-1}}, \ldots, y_{2^k-1}), & \text{если } x_k = 1, \end{cases}$$

притом что $f_1(0, y_0, y_1) = y_0$ и $f_1(1, y_0, y_1) = y_1$. Индукцией по $k \in \mathbb{N}$ докажем, что для f_k существует схема F_k размера не более $6 \cdot 2^k - 5$. Для $k = 1$ подходит схема

$$F_1(x_1, y_0, y_1) = (\neg x_1 \wedge y_0) \vee (x_1 \wedge y_1).$$

Действительно, $\mathrm{size}(F_1) = 7 \leqslant 6 \cdot 2 - 5$. Имея схему F_{k-1}, определяем F_k:

$$\begin{aligned}
F_k(x_1, \ldots, x_k, y_0, \ldots, y_{2^k-1}) &= (\neg x_k \wedge F_{k-1}(x_1, \ldots, x_{k-1}, y_0, \ldots, y_{2^{k-1}-1})) \vee \\
&\quad (x_k \wedge F_{k-1}(x_1, \ldots, x_{k-1}, y_{2^{k-1}}, \ldots, y_{2^k-1})).
\end{aligned}$$

Получаем

$$\mathrm{size}(F_k) = 2 \cdot \mathrm{size}(F_{k-1}) + 5 \leqslant 2(6 \cdot 2^{k-1} - 5) + 5 = 6 \cdot 2^k - 5.$$

Итак, $\mathrm{size}(F_k) \leqslant 6 \cdot 2^k - 5 \leqslant 6 \cdot 2^k = O(2^k)$. С другой стороны, $2^k \leqslant n$, так что $\mathrm{size}(F_k) = O(n)$. □

Задача 16.5. Докажите, что всякую функцию $f\colon \{0,1\}^n \to \{0,1\}$ можно вычислить булевой схемой размера $O(2^n)$.

Решение. Рассуждение с указательной функцией. Воспользуемся результатом задачи 16.4 — точнее, установленным нами усилением: размер схемы F_k имеет *линейно* ограниченный порядок роста относительно 2^k.

Заметим, что для произвольной $f\colon \{0,1\}^k \to \{0,1\}$ верно

$$f(x_1,\ldots,x_k) = f_k(x_k,\ldots,x_1, f(0\ldots00), f(0\ldots01), f(0\ldots10),\ldots, f(1\ldots10), f(1\ldots11)),$$

где значения f в правой части упорядочены по возрастанию чисел, двоичными записями которых являются соответствующие наборы. Таким образом, для построения вычисляющей $f(x_1,\ldots,x_k)$ схемы F нам понадобится схема F_k размера не более $6\cdot 2^k - 5$ и схема для вычисления (фиксированных для данной f) констант $f(0\ldots00), f(0\ldots01),\ldots, f(1\ldots11)$. Поскольку

$$\begin{aligned} 0 &= x_1 \wedge \neg x_1 \\ 1 &= x_1 \vee \neg x_1, \end{aligned}$$

размер схемы F_k достаточно увеличить на 3. Итак, $\mathrm{size}(F) = 6\cdot 2^k - 2 = O(2^k)$.

Прямое рассуждение. Легко проверить, что для любой функции $f\colon \{0,1\}^n \to \{0,1\}$ выполнено тождество

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

Докажем индукцией по n, что всякая функция f от n аргументов вычисляется некоторой схемой F размера не более $4\cdot 2^n - 5$. Для $n = 1$ утверждение проверяется непосредственно (рассмотрим все четыре функции одного аргумента). Предположим, что оно верно для $n \geqslant 1$, и рассмотрим случай $n+1$. По предположению, функции $f'(x_1,\ldots,x_n) = f(x_1,\ldots,x_n,0)$ и $f''(x_1,\ldots,x_n) = f(x_1,\ldots,x_n,1)$ вычисляются схемами F' и F'' соответственно, причем размер каждой схемы не превосходит $4\cdot 2^n - 5$. Согласно приведенном тождеству, функцию f вычисляет тогда схема

$$F(x_1,\ldots,x_n,x_{n+1}) = (\neg x_{n+1} \wedge F'(x_1,\ldots,x_n)) \vee (x_{n+1} \wedge F''(x_1,\ldots,x_n)),$$

причем, как видно,

$$\mathrm{size}(F) = \mathrm{size}(F') + \mathrm{size}(F'') + 5 \leqslant 2\cdot 4\cdot 2^n - 10 + 5 = 4\cdot 2^{n+1} - 5,$$

что и требовалось. Тем более, $\mathrm{size}(F) = O(2^n)$. □

Задача 16.6. Докажите, что для всех достаточно больших n существует монотонная булева функция $f\colon \{0,1\}^n \to \{0,1\}$, которую нельзя вычислить схемой размера меньше n^{100}. (Булева функция f называется монотонной, если из неравенств $x_i \leqslant y_i$ для всех i (систему которых кратко обозначим $\vec{x} \leqslant \vec{y}$) следует неравенство $f(x_1,\ldots,x_n) \leqslant f(y_1,\ldots,y_n)$.)

Решение. Оценим сверху число различных схем размера не более $s \geqslant n$. Как мы помним, схема (в стандартном базисе), есть последовательность функций, каждая из которых есть либо одна из переменных (которых у нас не более $n \leqslant s$ штук), либо

одна из базисных функций, примененная к предшествующим элементам схемы (не более чем к двум).

Чтобы закодировать номер каждого элемента — число от 1 до s — нам потребуется $\lfloor \log_2 s \rfloor + 1$ бит. Всю схему представим последовательностью длины s двоичных слов длины $3 + 2(\lfloor \log_2 s \rfloor + 1)$ («массивом записей»), где первые три бита кодируют тип элемента: переменная, отрицание, дизъюнкция, конъюнкция или «ничего» (схема может иметь длину менее s). Если элемент есть базисная функция, оставшиеся биты слова задают номера подставляемых в эту функцию предшествующих элементов. Если элемент есть переменная, эти биты содержат ее номер.

Указанное кодирование инъективно: разным схемам соответствуют разные последовательности. Поэтому схем размера $\leqslant s$ не больше, чем таких последовательностей, т. е. не более $2^{s(3+2(\lfloor \log_2 s \rfloor + 1))} \leqslant 2^{s(2s+5)} = O(2^{s^2})$. При $s = n^{100}$ получаем, что схем размера не более n^{100} будет $O(2^{n^{200}})$.

Покажем, с другой стороны, что мощность множества M_n монотонных функций растет в зависимости от n быстрее. Рассмотрим для данного n множество A двоичных наборов длины n, в которых имеется ровно $\lfloor \frac{n}{2} \rfloor$ единиц. Покажем, что каждое подмножество $N \subseteq A$ задает некоторую монотонную функцию f_N, так что отображение $N \mapsto f_N$ инъективно. Положим

$$f_N(\vec{\sigma}) = 1 \;\Leftrightarrow\; \text{существует } \vec{\tau} \in N,\ \text{т. ч. } \vec{\tau} \leqslant \vec{\sigma}.$$

В частности, $f_N(\vec{\sigma}) = 1$, если $\vec{\sigma} \in N$. С другой стороны, различные наборы из множества A попарно не сравнимы в смысле порядка $\leqslant$. Действительно, если $\vec{\sigma} < \vec{\tau}$ (это значит, по определению, $\vec{\sigma} \leqslant \vec{\tau}$ и $\vec{\sigma} \neq \vec{\tau}$), то в $\vec{\tau}$ строго больше единиц, чем в $\vec{\sigma}$. Поэтому $f_N(\vec{\sigma}) = 0$ при $\vec{\sigma} \in A \setminus N$. Следовательно, если $N_1 \neq N_2$, то и $f_{N_1} \neq f_{N_2}$.

Итак, инъективность установлена. Значит, различных функций f_N имеется столько же, сколько подмножеств в A, т. е. $2^{|A|} = 2^{\binom{n}{\lfloor \frac{n}{2} \rfloor}} = 2^{C(n)}$.

Остается убедиться, что $f_N \in M_n$. Допустим противное: пусть для наборов $\vec{\sigma} \leqslant \vec{\rho}$ неверно $f_N(\vec{\sigma}) \leqslant f_N(\vec{\rho})$, а значит, $f_N(\vec{\sigma}) = 1$ и $f_N(\vec{\rho}) = 0$. По определению f_N, найдется $\vec{\tau} \in N$, т. ч. $\vec{\tau} \leqslant \vec{\sigma}$. Но тогда, как легко заметит читатель, имеем $\vec{\tau} \leqslant \vec{\rho}$, откуда следует $f_N(\vec{\rho}) = 1$. Противоречие.

С учетом леммы 2, мы установили неравенство

$$2^{\frac{2^n}{n+1}} \leqslant 2^{C(n)} \leqslant |M_n|.$$

Для доказательства утверждения задачи остается заметить, что монотонных функций «больше», чем схем рассматриваемого размера, т. е. при любом выборе $C \in \mathbb{N}$ неравенство

$$C \cdot 2^{n^{200}} = 2^{n^{200} + \log_2 C} < 2^{\frac{2^n}{n+1}},$$

равносильное

$$(n^{200} + \log_2 C)(n+1) < 2^n,$$

будет выполнено при всех достаточно больших n. □

Задача 16.7. Постройте схемы полиномиального размера для деления n-битовых целых положительных чисел с остатком.

Решение. Располагая схемами полиномиального размера для умножения и вычитания, построенными на Лекции и в задаче 16.3, мы сможем вычислить остаток от деления, зная неполное частное. Сосредоточимся на вычислении последнего. Напомним, что неполное частное от деления (целого) числа $a \geqslant 0$ на число $b > 0$ есть число

$$q = \max\{p \in \mathbb{Z} \mid bp \leqslant a\},$$

где p в случае неотрицательных a и b можно считать лежащим от 0 до a. Обозначим неполное частое a/b. Из определения сразу вытекает, что если $1 + t = a/b$, то $t = (a - b)/b$. Отсюда $a/b = (a - bs)/b + s$ для всех $s \in \mathbb{N}$, т. ч. $bs \leqslant a$.

Простейший способ вычислить a/b — многократно вычитать b из a, пока разность остается неотрицательной. Кратность этой процедуры и составит a/b. Тем не менее, такой алгоритм имеет экспоненциальный порядок роста относительно *длины* входа. Так, из наибольшего n-разрядного числа $2^n - 1$ число 1 можно вычесть $2^n - 1$ раз.

Идея ускорения алгоритма состоит в том, чтобы вычитать достаточно большие числа и вычислять неполное частное поразрядно. Эта процедура аналогична школьному «делению уголком».

Итак, если $a < b$, то $a/b = 0$. Предположим, что $a \geqslant b$. Мы будем домножать b на 2^k при всевозможных k, т. ч. $b \cdot 2^k < 2^n$ (чтобы не выйти за n разрядов). Рассмотрим наибольшее подходящее k (а хотя бы одно найдется), при котором $b \cdot 2^k \leqslant a$ (отметим, что это условие влечет $b \cdot 2^k < 2^n$, так что мы выбросили заведомо негодные значения k). К оставшемуся числу $a - b \cdot 2^k < b \cdot 2^k$ применим ту же процедуру. Как видим, число $(a - b \cdot 2^k)/b < 2^k$ является k-разрядным. С другой стороны, по сделанному выше замечанию, $a/b = (a - b \cdot 2^k)/b + 2^k$, т.е. для определения a/b нужно приписать к $(a - b \cdot 2^k)/b$ единицу в старшем, $(k + 1)$-ом (нумеруя с 1-го) разряде.

Опишем, наконец, нашу схему. Схема LeadBit со входами $\vec{a}$, $\vec{b}$ при условии $a \geqslant b > 0$ вычисляет номер старшего разряда числа a/b. Точнее, она имеет выходы $z_n, \ldots, z_1$, для которых $z_k = 1$ тогда и только тогда, когда в числе a/b самая левая единица стоит в k-ом разряде, и, кроме того, выходы $w_n, \ldots, w_1$, представляющие число $a - b \cdot 2^k$. Значение $u_{k+1} = 1$ говорит, что b можно сдвигать на k разрядов влево без потери значащих цифр:

$$\begin{aligned}
1 &= b_1 \vee \neg b_1 \\
u_1 &= 1 \\
u_2 &= \neg b_n \\
u_3 &= \neg b_n \wedge \neg b_{n-1} \\
&\ldots \\
u_n &= \neg b_n \wedge \neg b_{n-1} \wedge \ldots \wedge \neg b_2
\end{aligned}$$

Теперь будем умножать b на 2^k, т. е. сдвигать b на k разрядов влево. Потери значащих цифр начнутся, когда $b \cdot 2^k$ превысит 2^{n-1}, но мы «обнулим» соответствующие

результаты с помощью $\vec{u}$. Результаты умножения вычтем из a:

$$\begin{array}{rcl}
0 & = & a_1 \wedge \neg a_1 \\
(x_n^1, \ldots, x_1^1, x_0^1) & = & \mathrm{Diff}_n(\vec{a}, b_n, \ldots, b_1) \\
(x_n^2, \ldots, x_1^2, x_0^2) & = & \mathrm{Diff}_n(\vec{a}, b_{n-1}, \ldots, b_1, 0) \\
(x_n^3, \ldots, x_1^3, x_0^3) & = & \mathrm{Diff}_n(\vec{a}, b_{n-2}, \ldots, b_1, 0, 0) \\
& \ldots & \\
(x_n^n, \ldots, x_1^n, x_0^n) & = & \mathrm{Diff}_n(\vec{a}, b_1, 0, \ldots, 0).
\end{array}$$

Проверяя знак разности, будем искать первый слева «плюс». При этом результаты «слишком больших» сдвигов будем учитывать, как со знаком «минус».

$$\begin{array}{rcl}
m_1 & = & x_0^1 \vee \neg u_1 \\
m_2 & = & x_0^2 \vee \neg u_2 \\
& \ldots & \\
m_n & = & x_0^n \vee \neg u_n \\
z_n & = & \neg m_n \\
z_{n-1} & = & m_n \wedge \neg m_{n-1} \\
z_{n-2} & = & m_n \wedge m_{n-1} \wedge \neg m_{n-2} \\
& \ldots & \\
z_1 & = & m_n \wedge m_{n-1} \wedge \ldots \wedge m_2 \wedge \neg m_1 \\
\vec{y}^{\,1} & = & \vec{x}^{\,1} \mathbin{\&\&} z_1 \\
\vec{y}^{\,2} & = & \vec{x}^{\,2} \mathbin{\&\&} z_2 \\
& \ldots & \\
\vec{y}^{\,n} & = & \vec{x}^{\,n} \mathbin{\&\&} z_n \\
\vec{y} & = & \vec{y}^{\,1} \mathbin{||} \vec{y}^{\,2} \mathbin{||} \ldots \mathbin{||} \vec{y}^{\,n} \\
(w_1, \ldots, w_n) & = & (y_1, \ldots, y_n)
\end{array}$$

Легко видеть, что схема LeadBit имеет размер $O(n^2)$. Итоговая схема по входам $\vec{a}$ и $\vec{b}$ возвращает n-разрядный выход $\vec{q}$:

$$\begin{array}{rcl}
l_n & = & L_n(\vec{a}, \vec{b}) \\
\vec{z}^{\,n} & = & (\mathrm{LeadBit}(\vec{a}, \vec{b}))_{\vec{z}} \mathbin{\&\&} \neg l_n \\
\vec{w}^{\,n} & = & (\mathrm{LeadBit}(\vec{a}, \vec{b}))_{\vec{w}} \\
l_{n-1} & = & L_n(\vec{w}^{\,n}, \vec{b}) \\
\vec{z}^{\,n-1} & = & (\mathrm{LeadBit}(\vec{w}^{\,n}, \vec{b}))_{\vec{z}} \mathbin{\&\&} \neg l_{n-1} \\
\vec{w}^{\,n-1} & = & (\mathrm{LeadBit}(\vec{w}^{\,n}, \vec{b}))_{\vec{w}} \\
& \ldots & \\
l_1 & = & L_n(\vec{w}^{\,2}, \vec{b}) \\
\vec{z}^{\,1} & = & (\mathrm{LeadBit}(\vec{w}^{\,2}, \vec{b}))_{\vec{z}} \mathbin{\&\&} \neg l_1 \\
\vec{q} & = & \vec{z}^{\,1} \mathbin{||} \vec{z}^{\,2} \mathbin{||} \ldots \mathbin{||} \vec{z}^{\,n}.
\end{array}$$

Очевидно, вся схема имеет размер $O(n^3)$. $\square$