# Дискретная математика. Модуль 3. Лекция 2

Лекторий ПМИ ФКН 2015-2016
Гринберг Вадим
Жижин Пётр
Пузырев Дмитрий

18 января 2016

## 1 Размер схемы. Сложность булевой функции. Верхние и нижние оценки сложности

### Размер схемы. Сложность булевой функции

*Размер булевой схемы* — это количество присваиваний в схеме $g_1, \ldots, g_L$ для вычисления функции $f : \{0,1\}^n \to \{0,1\}$.

*Сложность функции $f$ в базисе $B$* — это минимальный размер булевой схемы, вычисляющей функцию $f$ в базисе B. Если базис не указывают – имеют в виду стандартный базис $\{\neg, \vee, \wedge\}$. *Обозначение:* $C(f)$.

Утверждение: Если $B$ – конечный базис, тогда $\exists c : \forall f$ если $f$ вычисляется схемой $S$ в $B$ размера $L$, тогда существует схема в стандартном базисе, размера меньше, чем $c \cdot L$, вычисляющая ту же функцию $f$.

*Доказательство.* Пусть $f$ вычисляется схемой $S$ в базисе $B$ путём следующих присваиваний:

$$g_1, \ldots, g_k, \ldots, g_L = f$$

Рассмотрим некоторую $g_k$ как функцию в базисе $B$ от каких-то аргументов.

$$g_k = g(\ldots), g \in B$$

Для этой функции есть некоторая схема в стандартном базисе некоторого размера $L'_k$ (так как в стандартном базисе любая функция вычислима, в том числе и $g$).

Каждую $g_i$ заменим на соответствующую схему в стандартном базисе размера $L'_i$. Тогда и вся функция вычислима в стандартном базисе схемой размера:

$$L' = L'_1 + L'_2 + L'_3 + \ldots + L'_L \leqslant L \cdot max(L'_1, L'_2, \ldots L'_L) = L \cdot c_f$$

Для того чтобы теперь подобрать константу $c$ в определении опять возьмем наибольшее из всех $c_f$. **Q.E.D.**

### Верхняя оценка схемной сложности

**Теорема.** $C(f) = O(n \cdot 2^n)$

*Доказательство.* Повторим предыдущие рассуждения при доказательстве того, что в стандартном базисе любая функция вычислима. Для этого вспомним сокращенную ДНФ:

$$f(x) = \bigvee_{\substack{a : f(a)=1 \\ a \in \{0,1\}^n}} x^a, \; x^a = \bigwedge_{i=1}^{n} x_i^{a_i}$$

Нетрудно посчитать, что схема для вычисления $x^a$ имеет размер $L_a = O(n)$. Тогда итоговый размер схемы $L \leqslant 2^n \cdot O(n) \iff L = O(n \cdot 2^n)$. **Q.E.D.**

**Теорема.** $C(f) = O(\frac{2^n}{n})$

*Доказательство.* Доказательство предлагается почитать самостоятельно (например, в учебнике Вегенера). В нём достаточно много возни. Комментарий Михаила Николаевича:

> *Идея состоит в том, чтобы вычислить вообще все функции от k переменных, а потом использовать значения этих функций по многу раз в схеме вычисления данной функции f от n переменных. Оптимальный выбор k примерно логарифм n (с каким-то множителем). Более точно, зафиксировав набор значений n-k переменных, получаем функцию от k переменных. Ее мы насчитали на начальном этапе, а теперь уже просто используем. Всего будет $2^{n-k}$ обращений к таким схемам, какие-то будут использоваться чаще, какие-то реже - это зависит от функции f. У Вегенера написаны нужные рекуррентные формулы для этого рассуждения и дается оценка размера схемы, вычисляющей все функции от k переменных.*

**Q.E.D.**

## Нижняя оценка схемной сложности

**Теорема.** *Существует функция $f : \{0,1\}^n \to \{0,1\}$ такая, что $C(f) \geqslant \frac{2^n}{10n}$ (в точности то же самое, что $C(f) = \Omega(\frac{2^n}{n})$).*

*Доказательство.* Воспользуемся мощностным методом. Всего булевых функций от $n$ аргументов $2^{2^n}$.

Теперь узнаем, сколько булевых схем размера меньше либо равных некоторого фиксированного числа $L$. Для этого будем кодировать схемы двоичными словами. Посмотрим на какое-то присваивание в схеме $S$: $g_k = g(g_i, g_j)$. Для кодирования самой функции $g$ нужно 2 бита (так как в стандартном базисе всего три функции). Для кодирования номеров аргументов $i$ и $j$ нужно битов не более, чем $log_2 L$. А значит для всего присваивания $g_k$ нужно не более $2 \cdot (1 + log_2 L)$ бит.

Итого размер одной схемы в битах: $L \cdot 2 \cdot (1 + log_2 L)$. Каждая схема кодирует ровно одну функцию. А значит каждое двоичное слово кодирует не более одной функции (так как некоторые двоичные слова ни одну схему не задают).

Получается и схем размера $L$ не более, чем двоичных слов для схем такой длины, то есть: $2^{2L(1+log_2 L)}$.

Пусть $L = \frac{2^n}{10n}$. Размер схемы в битах тогда будет равен:

$$L_2 = \frac{2^n}{10n} \cdot 2 \cdot \left(1 + log_2\left(\frac{2^n}{10n}\right)\right) = \frac{2^n}{5n}\left(1 + n - log_2(10n)\right), 1 - log(10n) \leqslant 0 \implies L_2 \leqslant \frac{2^n}{5n} \cdot n = \frac{2^n}{5}$$

А значит функций, задающейся схемой такой длины не более чем $2^{\frac{2^n}{5}}$. Нетрудно заметить, что это число значительно меньше числа функций от $n$ аргументов.

$$2^{\frac{2^n}{5}} < 2^{2^n}$$

А значит существует функция, задающаяся схемой длины больше, чем $L$. **Q.E.D.**

Всё очень хорошо, но можно ли задать такую функцию явно? К сожалению, с этим есть некоторые проблемы. В 1984 году нашли $f$ такую, что $C(f) \geqslant 3n$. Прорывом 2015 стала функция $f$ такая, что $C(f) \geqslant \left(3 + \frac{1}{86}\right) n$.

## Схемы для сложения и умножения двоичных чисел. Схема для проверки графа на связность

Вспомним схему для сложения по модулю 2 из прошлой лекции. На каждом шаге добавлялось 5 присваиваний, значит, справедливо рекуррентное соотношение для количества операций $S_n = S_{n-1} + 5$. Из такого соотношения нетрудно сделать два вывода:

- $S_n$ вычисляется за O(n).
- Из того, что $S_1$ — константа, следует, что $S_n$ — также константа.

Изучим сложение двоичных чисел. Пусть у нас есть 2 двоичных числа $x : \{x_0, x_1, \ldots, x_{n-1}\}, y : \{y_0, y_1, \ldots, y_{n-1}\}$, где $x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{n-1}$ — двоичные разряды. Нужно выполнить схему $f : \{0,1\}^{2n} \to \{0,1\}^{n+1}$, результатом которой является $z : z_0, z_1, \ldots, z_n - 1$.

Вспомним привычный нам алгоритм сложения двоичных чисел поразрядно в столбик.

| $C_n$ | $C_{n-1}$ | $\ldots$ | $C_0$ |
|---|---|---|---|
| 0 | $x_{n-1}$ | $\ldots$ | $x_0$ |
| 0 | $y_{n-1}$ | $\ldots$ | $y_0$ |
| $z_n$ | $z_{n-1}$ | $\ldots$ | $z_0$ |

Обратим внимание на присутствие $C_0, \ldots, C_n$. Они являются тем числами (0 или 1) которые мы "запоминаем"при сложении и переносим на следующий разряд. Теперь рассмотрим отдельно некоторый $i$-й разряд сложения.

| $C_i$ |
|---|
| $x_i$ |
| $y_i$ |
| $z_i = x_i \oplus y_i \oplus c_i$ |

Нетрудно догадаться, что $z_i$ является суммой по модулю 2 $i$-х разрядов 2 слагаемых и "запомненного"числа от предыдущих разрядов. Возникает вопрос, как схемно выразить $c_{i+1}$ через предыдущие разряды? Оказывается, это также нетрудно сделать, воспользовавшись функцией MAJ: $c_{i+1} = MAJ(x_i, y_i, c_i)$.

Теперь поговорим о размере такой схемы. Как мы знаем из предыдущей лекции, сложение по модулю 2 и функцию большинства можно выполнить за константное число присваиваний. Получается, для операций $S_n, S_{n-1}$ справедливо рекуррентное соотношение $S_n = S_{n-1} + C$, где $C$ — некая константа. Из этого можно сделать вывод, что $S_n$ вычисляется за $O(n)$.

Теперь поговорим об умножении двоичных чисел. Также вспомним "школьную"схему умножения столбик. Как мы помним, нужно сначала посчитать результаты поочередного умножения разрядов $y$ на все разряды $x$:

$$u_i = y_i(x_0, x_1, \ldots, x_n)$$

Операция выполняется за $n \cdot O(n) = O(n^2)$.

Далее мы складываем получившиеся произведения $u_0 + u_1 + \ldots + u_n$. Операция выполнится за $O(n) \cdot O(2n) = O(n^2)$. Итого получаем выполнимость операции за $O(n^2)$.

Интересной задачей, выражаемой через схемы, является *проверка неориентированного графа на связность*: $Conn : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$.

Пусть булева переменная $x_{ij}, \{i,j\} \in F(G)$ принимает заначение 1 в том случае, если между вершинами $i$ и $j$ есть ребро. Рассмотрим функцию от таких булевых переменных.

Зададим *матрицу смежности* графа. Это матрица $A \in 0, 1^{n \times n}$, в которой на пересечении строки $i$ со столбцом $j$ стоит 1 тогда и только тогда, когда данные вершины связаны ребром. Такую матрицу и подадим на вход функции $Conn$. Заметим, что матрица смежности симметрична и на диагонали у нее обязательно стоят нули (мы запрещали петли — ребра, ведущие из вершины в нее же саму).

$$A = \begin{pmatrix} 0 & \dots & x_{1n} \\ x_{21} & \dots & x_{2n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & 0 \end{pmatrix}$$

Рассмотрим матрицу $A'$, которая отличается от матрицы $A$ тем, что у нее на главной диагонали стоят единицы, а не нули (в остальном матрицы совпадают).В терминах графов это означает, что к каждой вершине мы добавляем петлю. В модели простых неориентированных графов мы этого не допускали, но ничего не мешает нам рассмотреть графы с петлями. Идея состоит в том, что теперь, если между двумя вершинами есть путь длины меньше $n-1$, то есть и путь длины ровно $n-1$ (достаточно добавить к пути нужное количество петель). Нам достаточно взглянуть на $(A')^{n-1}$. Если в ячейках этой матрицы нет нулей, то граф связен, иначе не связен.

$$A' = \begin{pmatrix} 1 & \dots & x_{1n} \\ x_{21} & \dots & x_{2n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & 1 \end{pmatrix}$$

Также упрощение можно провести со способом возведения матрицы в степень. В данный момент мы это делаем над действительными числами, что заставляет нас складывать и умножать целые числа. Чтобы делать это с помощью схем, нам придется использовать описанные выше схемы для сложения и умножения, а чтобы оценить размер получившейся схемы придется оценивать величину возникающих в процессе вычислений целых чисел. Все это не очень хочется делать. Решение состоит в том, чтобы вместо умножения матриц над целыми числами воспользоваться так называемым *булевым умножением матриц.* В нем формулы для умножения матриц такие же, как и в обычном умножении, только вместо операции умножения используется конъюнкция, а вместо сложения — дизъюнкция. Тогда можно по индукции доказать, что в (булевой) матрице $A'^k$ на пересечении строки $i$ и столбца $j$ стоит 1 тогда и только тогда, когда в графе есть путь из вершины $i$ в вершину $j$ длины не больше $k$. Теперь мы готовы описать схему для проверки графа на связность. На вход схема (по существу) получает матрицу смежности $A'$. Схема последовательно вычисляет булевы степени этой матрицы $A'^2, \dots A'^{n-1}$. Затем схема вычисляет конъюнкцию всех ячеек матрицы $A'^{n-1}$ и подает ее на выход.

Оценим размер получившийся схемы. Для булева умножения двух булевых матриц размер $n \times n$ достаточно $n^2 \cdots O(n) = O(n^3)$ операций (каждая ячейка произведения матриц вычисляется за линейное число операций, всего ячеек $n^2$). Всего нам нужно $(n-1)$ умножение матриц, так что для вычисления матрицы $A'^{n-1}$ достаточно $O(n^4)$ операций. На последний этап (конъюнкция ячеек $A'^{n-1}$ нужно $O(n^2)$ операций, итого получается $O(n^4) + O(n^2) = O(n^4)$ операций.