Семинар по сложности булевых функций

Лекция 1: Введение

А. Куликов

Computer Science клуб при ПОМИ
http://compsciclub.ru

25.09.2011

План лекции

1. Булевы функции
2. Булевы схемы
3. Почти все функции имеют большую схемную сложность
   - Нижняя оценка
   - Верхняя оценка

Теория сложности вычислений

Структурная теория сложности изучает следующие вопросы: является ли память более мощным ресурсом, чем время? улучшают ли случайные числа мощь вычислительных ресурсов? легче ли проверить доказательство, чем найти его? Мы до сих пор не знаем ответов на эти вопросы. Большинство результатов в структурной теории сложности — условные, то есть опираются на недоказанные предположения (такие, как P≠NP).

Схемная сложность изучает нижние оценки на вычислительную сложность конкретных задач (например, умножение матриц или проверка наличия больших клик в графе). Рассматриваются конкретные модели вычисления — такие, как разрешающие деревья, ветвящиеся программы, булевы формулы, различные классы булевых схем, коммуникационные протоколы. Целью данной области являются безусловные нижние оценки.

Нижние оценки

- Простейшие булевы функции: конъюнкция (произведение) $x \cdot y$, исключающее ИЛИ (сумма по модулю 2) $x \oplus y$, ИЛИ $x \vee y$, отрицание $\neg x = 1 - x$.
- Центральная задача: сколько таких базовых операций необходимо для того, чтобы вычислить данную булеву функцию $f\colon \{0,1\}^n \to \{0,1\}$?
- Сложность доказательства нижних оценок идёт от нашего противника — схемы. Схемы небольшого размера могут производить вычисления очень контринтуитивно.
- Как доказать, что нет хитрого способа вычислить функцию?
- Данная задача лежит на стыке математики и computer science: нижние оценки очень важны для computer science, а их доказательства требуют методов комбинаторики, алгебры, мат. анализа и других областей математики.

План лекции

1. Булевы функции
2. Булевы схемы
3. Почти все функции имеют большую схемную сложность
   - Нижняя оценка
   - Верхняя оценка

Булевы функции

- B_n — это множество всех булевых функций $f\colon \{0,1\}^n \to \{0,1\}$ от n переменных.
- Булева функция $f \in B_n$ принимает вектор $a \in \{0,1\}^n$, если $f(a) = 1$, и отвергает в противном случае.
- Булева функция зависит от своей i-й переменной x_i, если найдутся такие константы $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$, что

$$f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n).$$

- Количество $|B_n|$ различных булевых функций от n переменных равно 2^{2^n}, то есть дважды экспоненциально.

Симметрические булевы функции

- Функция f называется симметрической, если её значение зависит только от суммы входных битов.
- Всего есть 2^{n+1} различных симметрических функций.

Примеры

- пороговая функция: $\mathrm{Th}_k^n(x) = 1 \Leftrightarrow x_1 + \cdots + x_n \geq k$;
- функция голосования: $\mathrm{Maj}_n(x) = 1 \Leftrightarrow x_1 + \cdots + x_n \geq \lceil n/2 \rceil$;
- функция чётности: $\oplus_n(x) = 1 \Leftrightarrow x_1 + \cdots + x_n \equiv 1 \pmod 2$;
- функция остатка по модулю: $\mathrm{MOD}_k^n = 1 \Leftrightarrow x_1 + \cdots + x_n \equiv 0 \pmod k$.

Ещё примеры булевых функций

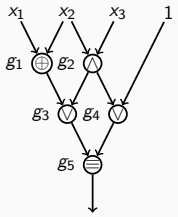
- В общем-то, любое свойство можно задать с помощью булевой функции.
- Например, свойство числа „быть простым“ задаёт булеву функцию PRIME: $\mathrm{PRIME}(x) = 1 \Leftrightarrow \sum_{i=1}^{n} x_i 2^{i-1}$ — простое число. Не так давно было доказано, что эта функция может быть детерминированно вычислена за полиномиальное от n время [Agrawal, Kayal, Saxena, 2004].
- Чтобы описать свойство графа, надо задать функцию на C_n^2 переменных. Каждый вектор x длины C_n^2 задает граф G_x: переменная x_{ij} отвечает за наличие ребра между вершинами i и j.
- Пример сложного для вычисления свойства графа — функция клики: $\mathrm{CLIQUE}(n,k)$ принимает вектор x тогда и только тогда, когда G_x содержит клику размера k.
- До сих пор не известно, может ли данная функция быть вычислена за полиномиальное время. Из доказательства отрицательного ответа будет тут же следовать P≠NP.

План лекции

1. Булевы функции
2. Булевы схемы
3. Почти все функции имеют большую схемную сложность
   - Нижняя оценка
   - Верхняя оценка

Пример схемы

$$
\begin{aligned}
g_1 &= x_1 \oplus x_2 \\
g_2 &= x_2 \wedge x_3 \\
g_3 &= g_1 \vee g_2 \\
g_4 &= g_2 \vee 1 \\
g_5 &= g_3 \equiv g4
\end{aligned}
$$



Формальное определение

- Итак, булева схема от n переменных над базисом Φ — это последовательность $g_1, \ldots, g_t$ из $t \geq n$ булевых функций, такая что первые n функций являются просто входными переменными $g_1 = x_1, \ldots, g_n = x_n$, а каждая следующая функция g_i является применением $g_i = \phi(g_{i_1}, \ldots, g_{u_d})$ базовой функции $\phi \in \Phi$ к некоторым из предыдущих функций.
- Размером схемы называется количество $(t - n)$ её гейтов (функциональных элементов). Глубиной называется длина самого длинного пути от входной переменной до выхода.

Формулы и схемы де Моргана

- Формулой называется схема, соответствующий граф которой является деревом. Основное отличие от схем заключается в том, что результат вычислений каждого гейта не может быть использован более одного раза. Формула может быть записана в одну строчку.
- Схемой де Моргана называется схема над базисом $\{\wedge, \vee\}$, входами которой являются литералы, то есть переменные и их отрицания. Другими словами, отрицания применяются только ко входам. Используя правила де Моргана $\neg(x \vee y) = \neg x \wedge \neg y$ и $\neg(x \wedge y) = \neg x \vee \neg y$, любую схему над базисом $\{\wedge, \vee, \neg\}$ можно преобразовать в схему де Моргана, увеличив размер не более, чем вдвое.

План лекции

1. Булевы функции
2. Булевы схемы
3. Почти все функции имеют большую схемную сложность
   - Нижняя оценка
   - Верхняя оценка

План лекции

1. Булевы функции
2. Булевы схемы
3. Почти все функции имеют большую схемную сложность
   - Нижняя оценка
   - Верхняя оценка

Функция Шэннона

- Функция Шэннона (для некоторой модели схем): $\mu(n)$ — это максимальный размер схемной сложности функции от n переменных. Другими словами, $\mu(n)$ есть такое минимальное число t, что любая функция от n переменных может быть вычислена схемами размера t.
- Как правило, под функцией f подразумевается бесконечная последовательность функций $f = \{f_n \mid n = 1, 2, \ldots\}$.
- Замечание: в дальнейшем рассматриваем схемы над полным бинарным базисом B_2.

Схемная сложность случайной функции

- Из соображений мощности (Shannon, 1949): оценим, какое количество функций от n переменных могут быть вычислены схемам размера t, и сравним полученное число с числом 2^{2^n} всех функций от n переменных.
- Число $F(n, t)$ схем размера $\leq t$ от n переменных не превосходит
$$\left(16(t + n + 2)^2\right)^t.$$
Каждый из t гейтов вычисляет одну из 16 возможных бинарных булевых функций и зависит от двух предыдущих гейтов, которые могут быть гейтов ($\leq t$ возможностей) или же переменной или константой ($\leq n + 2$ возможностей).
- Для $t = 2^n/(10n)$, $F(n, t)$ приблизительно равно $2^{2^n/5}$, что $\ll 2^{2^n}$.
- Таким образом, схемная сложность почти всех функций от n переменных экспоненциальна по n. В то же время не известно ни одной явной функции, требующей схем более чем линейного размера.

План лекции

1. Булевы функции
2. Булевы схемы
3. Почти все функции имеют большую схемную сложность
   - Нижняя оценка
   - Верхняя оценка

Представление функции многочленом над GF(2)

- Любую функцию $f \in B_n$ можно представить (причём единственным образом) многочленом над GF(2).
- Чтобы представить функцию многочленом, достаточно взять все вектора, принимаемые этой функцией, и сложить соответствующие им элементарные конъюнкции.

Пример: Thr_2^3

- Рассмотрим функцию Thr_2^3: $\mathrm{Thr}_2^3(x_1, x_2, x_3) = 1 \Leftrightarrow x_1 + x_2 + x_3 \geq 2$.
- Соответствующий многочлен: $x_1x_2x_3 + (1 - x_1)x_2x_3 + x_1(1 - x_2)x_3 + x_1x_2(1 - x_3) = x_1x_2 + x_2x_3 + x_3x_1$.

- Итак, любая функция представляется многочленом. Чтобы понять, что такой многочлен единственен, достаточно заметить, что количество различных функций и количество различных многочленов совпадают и равны 2^{2^n}.

Очевидная верхняя оценка

- Из представления функции многочленом сразу следует, что схемная сложность любой функции не превосходит $n2^n$.
- Такую же верхнюю оценку можно получить, заметив, что любую функцию можно представить в виде конъюнктивной нормальной формы (КНФ), то есть конъюнкции дизъюнкций литералов, или конъюнктивной нормальной формы (КНФ), то есть дизъюнкции конъюнкций литералов.

Пример: ДНФ для Thr_2^3

$x_1x_2x_3 \vee \bar{x_1}x_2x_3 \vee x_1\bar{x_2}x_3 \vee x_1x_2\bar{x_3}$

Улучшение

- Чтобы получить более сильную оценку, нужно заметить, что функцию $f \in B_n$ всегда можно представить как

$$f(x_1, \ldots, x_n) = f_0(x_1, \ldots, x_{n-1}) + x_n f_\Delta(x_1, \ldots, x_{n-1}),$$

где $f_\Delta = f_0 + f_1$, а $f_0 = f\,|_{x_n=0}$ и $f_1 = f\,|_{x_n=1}$.

- Это даёт нам следующее рекуррентное соотношение:

$$\mu(n) \leq 2\mu(n-1) + 2,\ \mu(1) = 1.$$

- Из данного соотношения следует верхняя оценка $\mu(n) \leq \frac{3}{4}2^n - 2$.

Дальнейшее улучшение

- Идея улучшения (Shannon, 1949): вместо того, чтобы рекурсивно спускаться до $n = 1$, предвычислим все булевы функции от $k \ll n$ переменных.
- Это даст нам рекуррентное соотношение вида $\mu'(n) \leq 2\mu'(n-1) + 2,\ \mu'(k) = 0$.
- Тогда после предвычисления нам понадобится не более $\mu'(n) \leq 2 \cdot 2^{n-k} - 2$ гейтов.
- Ниже мы покажем, что на предвычисление понадобится не более 2^{2^k} гейтов.
- Тогда $\mu(n) \leq 2^{2^k} + 2 \cdot 2^{n-k} - 2$.
- Взяв $k = \lfloor \log_2 n - \epsilon \rfloor$, получим, что $\mu(n) \leq 2^{2+\epsilon} \cdot 2^n/n + o(2^n/n)$.

Вычисление всех функций одновременно

- Итак, осталось показать, как вычислить все функции от k переменных за 2^{2^k} гейтов.
- Сначала вычислим все мономы, последовательно увеличивая их размер. Потом вычислим все суммы мономов, последовательно увеличивая количество членов в сумме. Таким образом, получим все 2^{2^k} многочленов от k переменных.
- Мономы степени 0 и 1 у нас уже есть — это просто две константы и k входных переменных.
- Чтобы получить моном степени d, достаточно одного гейта: нужно взять моном степени $d - 1$ и домножить его на переменную.
- Чтобы получить сумму s мономов, достаточно одного гейта: нужно взять сумму $(s - 1)$ мономов и прибавить к ней моном.

Более точная оценка

Используя более хитрую конструкцию, Лупанов (1958) доказал, что

$$\mu(n) \leq \left(1 + O\left(\frac{\log n}{n}\right)\right)\frac{2^n}{n}.$$

Текущие рекорды

- Итак, мы узнали, что схемная сложность почти любой функции из B_n есть $\Theta(2^n/n)$.
- Лучшие доказанные на данный момент оценки для явных функций:

	схемы	формулы
полный бинарный базис B_2	$3n - o(n)$ [Blum]	$n^{2-o(1)}$ [Нечипорук]
базис $U_2 = B_2 \setminus \{\oplus, \equiv\}$	$5n - o(n)$ [Iwama et al.]	$n^{3-o(1)}$ [Hastad]
монотонный базис $M_2 = \{\vee, \wedge\}$	экспоненциальная [Разборов; Alon, Boppana; Андреев; Karchmer, Wigderson]	

Упражнения

- Докажите, что для любой функции от n переменных есть вычисляющая её формула де Моргана, в которой не более $4 \cdot 2^n - 2$ листьев.
- Пусть $m = \lceil \log_2(n+1) \rceil$ и пусть $\mathrm{Sum}_n \colon \{0,1\}^n \to \{0,1\}^m$ по входу x выдаёт бинарное представление суммы его битов. Рассмотрим схемы над полным бинарным базисом.
  - Покажите, что $C(\mathrm{Sum}_3) \leq 5$.
  - Покажите, что $C(\mathrm{Sum}_n) \leq 5n$.
  - Покажите, что $C(f_n) \leq 5n + o(n)$, где f_n — симметрическая функция от n переменных.
- (Схемы как линейные программы.) Пусть $F(x)$ — схема над $\{\vee, \wedge, \neg\}$ с m гейтами. Покажите, что найдётся система $L(x, y)$ из $O(m)$ линейных неравенств с коэффициентами ± 1, такая что для всех $x \in \{0,1\}^n$, $F(x) = 1$ тогда и только тогда, когда найдётся такой вектор y, что все неравенства системы $L(x, y)$ выполнены.

Спасибо за внимание!