Структурная теория сложности

Эдуард Алексеевич Гирш

http://logic.pdmi.ras.ru/~hirsch

ПОМИ РАН

26 октября 2008 г.

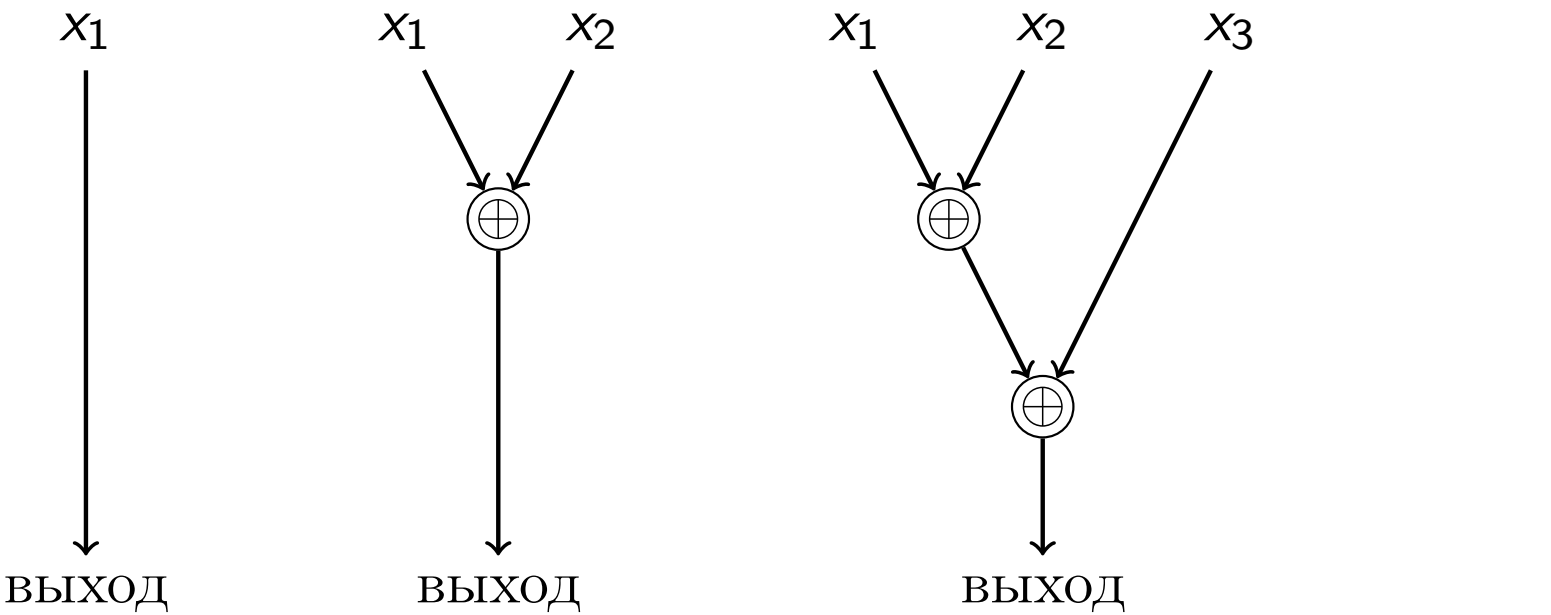
Напоминание: полиномиальные схемы

$L \in \mathbf{Size}[f(n)]$, если существует семейство булевых схем $\{C_n\}_{n\in\mathbb{N}}$, т.ч.

- $\forall n\ |C_n| \le f(n)$,
- $\forall x\ (x \in L \Leftrightarrow C_{|x|}(x) = 1)$.

Полиномиальные схемы:

$$\mathbf{P/poly} = \bigcup_{k\in\mathbb{N}} \mathbf{Size}[n^k].$$



Теорема Карпа-Липтона

Теорема

NP $\subseteq$ **P**/**poly** $\Rightarrow$ **PH** $=$ Σ^2**P**.

Покажем, что Σ^3**P**-полный язык

$$\mathtt{QBF}_3 = \{F - \text{формула в КНФ} \mid \exists x \forall y \exists z \; F(x, y, z)\}.$$

лежит в Σ^2**P**.

Теорема Карпа-Липтона

Теорема

NP $\subseteq$ **P**/**poly** $\Rightarrow$ **PH** $=$ Σ^2**P**.

Покажем, что Σ^3**P**-полный язык

$$\mathtt{QBF}_3 = \{F \text{ — формула в КНФ} \mid \exists x \forall y \exists z\ F(x, y, z)\}.$$

лежит в Σ^2**P**.

Проверка корректности схем для `SAT`:

$$C_{|G|}(G) \stackrel{?}{=} C_{|G[x_1:=0]|}(G[x_1 := 0]) \vee C_{|G[x_1:=1]|}(G[x_1 := 1])$$

и проверка корректности для тривиальных формул.

Теорема Карпа-Липтона

Теорема

NP $\subseteq$ **P**/**poly** $\Rightarrow$ **PH** $=$ Σ^2**P**.

Покажем, что Σ^3**P**-полный язык

$$\mathtt{QBF}_3 = \{F \text{ — формула в КНФ} \mid \exists x \forall y \exists z \; F(x, y, z)\}.$$

лежит в Σ^2**P**.

$(\exists x \forall y \exists z (F)) \in \mathtt{QBF}_3 \Leftrightarrow$
$\exists$ схемы $C_1, \ldots, C_{|F|}$
$\exists x$
$\forall y$
$\forall G$ — булевой формулы длины $\leq |F|$
(семейство $\{C_i\}$ корректно для G) $\wedge$ $C_{|F|}(F(x, y, z)) = 1$.

Схемы фиксированного полиномиального размера

Теорема

$\forall k \;\; \Sigma^4\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$.

Из соображений мощности имеется $f : \{0,1\}^n \to \{0,1\}$, зависящая от первых $c \cdot k \cdot \log n$ битов, для которой нет булевой схемы размера n^k.

Сконструируем такую функцию в $\Sigma^4\mathbf{P}$.

$$y \in L \iff \exists f \; \forall c \; (\text{схемы размера } n^k) \qquad \exists x \qquad :$$
$$\underbrace{f(x) \neq c(x)}_{\text{не принимается схемой}} \qquad .$$

Схемы фиксированного полиномиального размера

Теорема

$\forall k \;\; \Sigma^4\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$.

Из соображений мощности имеется $f : \{0,1\}^n \to \{0,1\}$, зависящая от первых $c \cdot k \cdot \log n$ битов, для которой нет булевой схемы размера n^k.

Сконструируем такую функцию в $\Sigma^4\mathbf{P}$.

$$y \in L \iff \exists f \; \forall c \; (\text{схемы размера } n^k) \qquad \exists x \qquad :$$
$$\underbrace{f(x) \neq c(x)}_{\text{не принимается схемой}} \qquad \wedge \; \underbrace{f(y) = 1}_{\text{значение}}.$$

Схемы фиксированного полиномиального размера

Теорема

$\forall k \;\; \Sigma^4\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$.

Из соображений мощности имеется $f : \{0,1\}^n \to \{0,1\}$, зависящая от первых $c \cdot k \cdot \log n$ битов, для которой нет булевой схемы размера n^k.

Сконструируем такую функцию в $\Sigma^4\mathbf{P}$.

$$y \in L \iff \exists f \; \forall c \; (\text{схемы размера } n^k) \; \forall f' \; \exists x \exists c' \; (\text{схема}\dots) \;\; \forall x' : \underbrace{f(x) \neq c(x)}_{\text{не принимается схемой}} \;\; \wedge \; \underbrace{((f \leq f') \vee f'(x') = c'(x'))}_{\text{первая такая } f} \;\; \wedge \; \underbrace{f(y) = 1}_{\text{значение}}.$$

Схемы фиксированного полиномиального размера

Теорема

$\forall k \;\; \Sigma^4\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$.

Из соображений мощности имеется $f : \{0,1\}^n \to \{0,1\}$, зависящая от первых $c \cdot k \cdot \log n$ битов, для которой нет булевой схемы размера n^k.

Сконструируем такую функцию в $\Sigma^4\mathbf{P}$.

$$y \in L \iff \exists f \; \forall c \; (\text{схемы размера } n^k) \; \forall f' \; \exists x \exists c' \; (\text{схема}\dots) \;\; \forall x' : \underbrace{f(x) \neq c(x)}_{\text{не принимается схемой}} \;\; \wedge \; \underbrace{((f \leq f') \vee f'(x') = c'(x'))}_{\text{первая такая } f} \;\; \wedge \; \underbrace{f(y) = 1}_{\text{значение}}.$$

Остаётся превратить n^k в $O(n^k)$.

Схемы фиксированного полиномиального размера

Теорема

$\forall k \quad \Sigma^4\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$.

Следствие

$\forall k \quad \Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$.

$\Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P} \subseteq \mathbf{Size}[n^k] \Rightarrow$
$\mathbf{NP} \subseteq \mathbf{P/poly} \Rightarrow$
$\mathbf{PH} = \Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P} \subseteq \mathbf{Size}[n^k]$.

Равномерные полиномиальные схемы

. . . и параллельные вычисления

Семейство схем $\{C_n\}_{n\in\mathbb{N}}$ равномерно, если имеется полиномиальный алгоритм A, т.ч. $A(1^n) = C_n$.

Ясно, что равномерные полиномиальные схемы задают $\mathbf{P}$.

Logspace-равномерные: A использует память $O(\log n)$.

Равномерные полиномиальные схемы

. . . и параллельные вычисления

Семейство схем $\{C_n\}_{n\in\mathbb{N}}$ равномерно, если имеется полиномиальный алгоритм A, т.ч. $A(1^n) = C_n$.

Ясно, что равномерные полиномиальные схемы задают $\mathbf{P}$.

Logspace-равномерные: A использует память $O(\log n)$.

Глубина схемы $\sim$ время параллельного вычисления (см. доску).

$$\mathbf{NC}^i = \left\{ L \;\middle|\; \begin{array}{l} \text{для } L \text{ есть logspace-равномерные} \\ \text{схемы глубины } O(\log^i n) \end{array} \right\}.$$

$$\mathbf{NC} = \bigcup_i \mathbf{NC}^i \qquad \subseteq \mathbf{P}.$$

P-полнота

Сведéния с $O(\log n)$ памяти.

Теорема

Если L — **P**-полный, то

- $L \in \mathbf{NC} \iff \mathbf{P} = \mathbf{NC}$ (всё параллелизуется);
- $L \in \mathbf{DSpace}[\log] \iff \mathbf{P} = \mathbf{DSpace}[\log]$.

P-полный язык:

$$\text{CIRCUIT\_EVAL} = \{(\text{схема } C, \text{вход } x) \mid C(x) = 1\}.$$

NSpace

$$\mathbf{NSpace}[f(n)] = \{L \mid L \text{ принимается НМТ с памятью } O(f(n))\}.$$

$f(n)$ должна быть неубывающей и вычислимой с памятью $O(f(n))$ по 1^n.

В определении с подсказкой лента подсказки читается слева направо!

$$\mathbf{NPSPACE} = \bigcup_{k \geq 0} \mathbf{NSpace}[n^k].$$

STCON; **NSpace** vs **DSpace**

$$\texttt{STCON} = \{(G, s, t) \mid G \text{ — ор.граф, } s \rightsquigarrow t\}.$$

Лемма

$\texttt{STCON} \in \mathbf{DSpace}[\log^2 n]$.

$$\begin{aligned} \mathrm{PATH}(x, y, i) &= \exists \text{ путь из } x \text{ в } y \text{ длины не более } 2^i. \\ \mathrm{PATH}(x, y, i) &= \bigvee_z (\mathrm{PATH}(x, z, i-1) \wedge \mathrm{PATH}(z, y, i-1)). \end{aligned}$$

Теорема

$\mathbf{NSpace}(f) \subseteq \mathbf{DSpace}(f^2)$ для $f(n) = \Omega(\log n)$.

Следствие

$\mathbf{PSPACE} = \mathbf{NPSPACE}$.

STCON; **NSpace** vs **DSpace**

$$\texttt{STCON} = \{(G, s, t) \mid G \text{ — ор.граф, } s \rightsquigarrow t\}.$$

Лемма

$\texttt{STCON} \in \mathbf{DSpace}[\log^2 n]$.

$$\begin{aligned} \mathrm{PATH}(x, y, i) &= \exists \text{ путь из } x \text{ в } y \text{ длины не более } 2^i. \\ \mathrm{PATH}(x, y, i) &= \bigvee_z (\mathrm{PATH}(x, z, i-1) \wedge \mathrm{PATH}(z, y, i-1)). \end{aligned}$$

Лемма

$\texttt{STCON}$ является $\mathbf{NSpace}[\log n]$-полной.

STCON; **NSpace** vs **DSpace**

$$\texttt{STCON} = \{(G, s, t) \mid G \text{ — ор.граф, } s \rightsquigarrow t\}.$$

Лемма

$\texttt{STCON} \in \mathbf{DSpace}[\log^2 n]$.

$$\begin{aligned} \mathrm{PATH}(x, y, i) &= \exists \text{ путь из } x \text{ в } y \text{ длины не более } 2^i. \\ \mathrm{PATH}(x, y, i) &= \bigvee_z (\mathrm{PATH}(x, z, i-1) \wedge \mathrm{PATH}(z, y, i-1)). \end{aligned}$$

Лемма

$\texttt{STCON}$ является $\mathbf{NSpace}[\log n]$-полной.

Факт: для неор.графов: $\texttt{USTCON} \in \mathbf{DSpace}[\log n]$ [Reingold, 2004].

Вопрос на засыпку: а кто $\mathbf{DSpace}[\log n]$-полная?

Параллельные вычисления и логарифмическая память

Теорема

$\mathbf{NC}^1 \subseteq \mathbf{DSpace}(\log) \subseteq \mathbf{NSpace}(\log) \subseteq \mathbf{NC}^2$.

Параллельные вычисления и логарифмическая память

Теорема

$\mathbf{NC}^1 \subseteq \mathbf{DSpace}(\log) \subseteq \mathbf{NSpace}(\log) \subseteq \mathbf{NC}^2$.

Пусть logspace НМТ M принимает $L \in \mathbf{NSpace}(\log)$.

- Интересует достижимость в графе конфигураций M.
- Для конкретной входной ленты их полиномиальное число k.

Параллельные вычисления и логарифмическая память

Теорема

$\mathbf{NC}^1 \subseteq \mathbf{DSpace}(\log) \subseteq \mathbf{NSpace}(\log) \subseteq \mathbf{NC}^2$.

Пусть logspace НМТ M принимает $L \in \mathbf{NSpace}(\log)$.

- Интересует достижимость в графе конфигураций M.
- Для конкретной входной ленты их полиномиальное число k.
- A — матрица смежности $(k \times k)$.
- Достаточно вычислить A^k.

Параллельные вычисления и логарифмическая память

Теорема

$\mathbf{NC}^1 \subseteq \mathbf{DSpace}(\log) \subseteq \mathbf{NSpace}(\log) \subseteq \mathbf{NC}^2$.

Пусть logspace НМТ M принимает $L \in \mathbf{NSpace}(\log)$.

- Интересует достижимость в графе конфигураций M.
- Для конкретной входной ленты их полиномиальное число k.
- A — матрица смежности $(k \times k)$.
- Достаточно вычислить A^k.
- Это $\log k$ последовательных умножений: A^2, $(A^2)^2$, ...
- Умножение пары булевых матриц: схема глубины $O(\log k)$, см. на доску.

Параллельные вычисления и логарифмическая память

Теорема

$\mathbf{NC}^1 \subseteq \mathbf{DSpace}(\log) \subseteq \mathbf{NSpace}(\log) \subseteq \mathbf{NC}^2$.

Лемма. Композиция двух logspace функций $f_2(f_1(x))$.

- Сделать выходную ленту f_1 входной лентой f_2 нельзя.
- Храним только счётчики позиций.
- Нужен очередной бит входа f_2 — продолжим работу f_1.
  (Если лента не write-once, можно доводить до конца каждый раз.)

Параллельные вычисления и логарифмическая память

Теорема

$\mathbf{NC}^1 \subseteq \mathbf{DSpace}(\log) \subseteq \mathbf{NSpace}(\log) \subseteq \mathbf{NC}^2$.

$x \overset{?}{\in} L \in \mathbf{NC}^1$. Строим композицию трёх logspace функций с логарифмической памятью.

1. Строим схему (семейство было logspace-равномерным).
2. Преобразуем схему в формулу (dag $\longrightarrow$ дерево):
   - гейт $\longrightarrow$ путь от выхода (битовая строка),
   - поиск в глубину — логарифмическая память,
   - для возврата идём заново от корня.
3. Вычисляем значение формулы на входе x.
   - Снова поиск в глубину.