Введение в теорию сложности вычислений

Эдуард Алексеевич Гирш

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

СПбГУ и ПОМИ РАН

лекция 26 ноября 2020 г.

$\Sigma^k\mathbf{P}$-полная задача: $\mathtt{QBF}_k$

Язык $\mathtt{QBF}_k$ состоит из замкнутых истинных формул вида

$$\exists X_1 \forall X_2 \exists X_3 \ldots X_k\, \phi,$$

$R(\exists \ldots \phi, \bar{x}_1, \bar{x}_2 \ldots)$

где ϕ — формула в КНФ или ДНФ, а $\{X_i\}_{i=1}^k$ — разбиение множества переменных этой формулы (на непустые непересекающиеся подмножества).

Следствие

$\mathtt{QBF}_k$ — $\Sigma^k\mathbf{P}$-полна.

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное $R \in P$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots R(x, y_1, y_2, \ldots, y_k))$.
Если последний квантор — $\exists$, то запишем R в виде булевой формулы Φ как в теореме Кука-Левина:

$$R(z) \Leftrightarrow \exists w \Phi(z, w),$$

итого:

$$\forall x\ (x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots \exists y_k \exists w \Phi(x, y_1, y_2, \ldots, y_k, w)).$$

$\Sigma^k\mathbf{P}$-полная задача: $\mathtt{QBF}_k$

Язык $\mathtt{QBF}_k$ состоит из замкнутых истинных формул вида

$$\exists X_1 \forall X_2 \exists X_3 \ldots X_k\, \phi,$$

$X_1 = x_{11}x_{12}\cdots$

где ϕ — формула в КНФ или ДНФ, а $\{X_i\}_{i=1}^k$ — разбиение множества переменных этой формулы (на непустые непересекающиеся подмножества).

Следствие

$\mathtt{QBF}_k$ — $\Sigma^k\mathbf{P}$-полна.

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное $R \in P$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots R(x, y_1, y_2, \ldots, y_k))$.
Если последний квантор — $\forall$, то запишем $\overline{R}$ в виде булевой формулы Ψ как в теореме Кука-Левина:

$$\overline{R}(z) \Leftrightarrow \exists w \Psi(z, w),$$

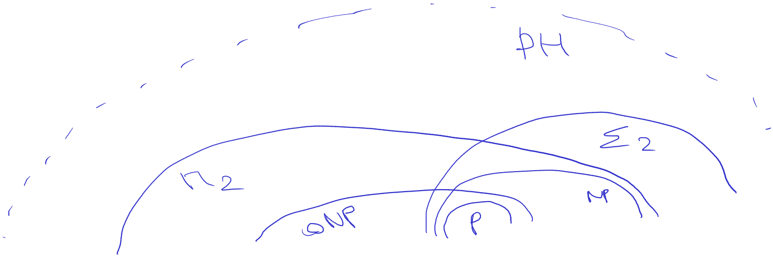
итого:

$$\forall x\ (x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots \underbrace{\forall y_k \forall w}\overline{\Psi}(x, y_1, y_2, \ldots, y_k, w)).$$

Коллапс полиномиальной иерархии

Теорема

Если $\Sigma^k\mathbf{P} = \Pi^k\mathbf{P}$, то $\mathbf{PH} = \Sigma^k\mathbf{P}$.



Коллапс полиномиальной иерархии

Теорема

Если $\Sigma^k\mathbf{P} = \Pi^k\mathbf{P}$, то $\mathbf{PH} = \Sigma^k\mathbf{P}$.

Достаточно показать $\Sigma^{k+1}\mathbf{P} = \Pi^k\mathbf{P}$.

Пусть $L \in \Sigma^{k+1}\mathbf{P}$, т.е. $L = \{x : \exists y\, R(x, y)\}$ для $R \in \Pi^k\mathbf{P} = \Sigma^k\mathbf{P}$.

Значит, имеется $S \in \Pi^{k-1}\mathbf{P}$, т.ч. $R(x, y) \Leftrightarrow \exists z\, S(x, y, z)$, т.е.
$x \in L \Leftrightarrow \exists y \exists z\, S(x, y, z)$, т.е. $L \in \Sigma^k\mathbf{P}$.

Коллапс полиномиальной иерархии

Теорема

Если $\Sigma^k \mathbf{P} = \Pi^k \mathbf{P}$, то $\mathbf{PH} = \Sigma^k \mathbf{P}$.

Достаточно показать $\Sigma^{k+1}\mathbf{P} = \Pi^k \mathbf{P}$.

Пусть $L \in \Sigma^{k+1}\mathbf{P}$, т.е. $L = \{x : \exists y\, R(x, y)\}$ для $R \in \Pi^k \mathbf{P} = \Sigma^k \mathbf{P}$.

Значит, имеется $S \in \Pi^{k-1}\mathbf{P}$, т.ч. $R(x, y) \Leftrightarrow \exists z\, S(x, y, z)$, т.е.
$x \in L \Leftrightarrow \exists y \exists z\, S(x, y, z)$, т.е. $L \in \Sigma^k \mathbf{P}$.

Коллапс полиномиальной иерархии

Теорема

Если $\Sigma^k\mathbf{P} = \Pi^k\mathbf{P}$, то $\mathbf{PH} = \Sigma^k\mathbf{P}$.

Достаточно показать $\Sigma^{k+1}\mathbf{P} = \Pi^k\mathbf{P}$.

Пусть $L \in \Sigma^{k+1}\mathbf{P}$, т.е. $L = \{x : \exists y\, R(x, y)\}$ для $R \in \Pi^k\mathbf{P} = \Sigma^k\mathbf{P}$.

Значит, имеется $S \in \Pi^{k-1}\mathbf{P}$, т.ч. $R(x, y) \Leftrightarrow \exists z\, S(x, y, z)$, т.е.
$x \in L \Leftrightarrow \exists y \exists z\, S(x, y, z)$, т.е. $L \in \Sigma^k\mathbf{P}$.

Следствие

Если существует **PH**-полная задача, то полиномиальная иерархия конечна.

Коллапс полиномиальной иерархии

Теорема

Если $\Sigma^k \mathbf{P} = \Pi^k \mathbf{P}$, то $\mathbf{PH} = \Sigma^k \mathbf{P}$.

Достаточно показать $\Sigma^{k+1}\mathbf{P} = \Pi^k \mathbf{P}$.

Пусть $L \in \Sigma^{k+1}\mathbf{P}$, т.е. $L = \{x : \exists y\, R(x, y)\}$ для $R \in \Pi^k \mathbf{P} = \Sigma^k \mathbf{P}$.

Значит, имеется $S \in \Pi^{k-1}\mathbf{P}$, т.ч. $R(x, y) \Leftrightarrow \exists z\, S(x, y, z)$, т.е.
$x \in L \Leftrightarrow \exists y \exists z\, S(x, y, z)$, т.е. $L \in \Sigma^k \mathbf{P}$.

Следствие

Если существует $\mathbf{PH}$-полная задача, то полиномиальная иерархия конечна.

Полный язык ведь лежит в конкретном $\Sigma^k \mathbf{P}$.

Σ^{k+1}

$L \in \Sigma^k$

PH

Классы, ограниченные по времени и памяти

$\mathbf{DTime}[f(n)] = \{L \mid L$ принимается ДМТ, работающей время $O(f(n))\}$.

$f(n)$ должна быть неубывающей и вычислимой за время $O(f(n))$ по 1^n.

$$\mathbf{P} = \bigcup_{k \geq 0} \mathbf{DTime}[n^k].$$

$\mathbf{DSpace}[f(n)] = \{L \mid L$ принимается ДМТ с памятью $O(f(n))\}$.

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

$$\mathbf{PSPACE} = \bigcup_{k \geq 0} \mathbf{DSpace}[n^k].$$

PSPACE-полная задача: QBF

Язык QBF состоит из замкнутых истинных формул вида

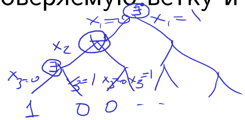
$$q_1 x_1 q_2 x_2 \ldots \phi,$$

где ϕ — формула в КНФ, $q_i = \forall$ или $q_i = \exists$.

Теорема

QBF **PSPACE**-полна.

QBF $\in$ **PSPACE**:
Рассмотрим дерево перебора всех значений переменных.
В каждом листе запишем значение (бескванторной) формулы.
Рекурсивно, поиском в глубину вычислим результат.
Хранить нужно лишь проверяемую ветку и два последних значения.



$x_1 = 0$
$x_2 = 1$
...
$\Phi(x_1 = 0, x_2 = 1, \ldots)$

PSPACE-полная задача: QBF

Язык QBF состоит из замкнутых истинных формул вида

$$q_1 x_1 q_2 x_2 \ldots \phi,$$

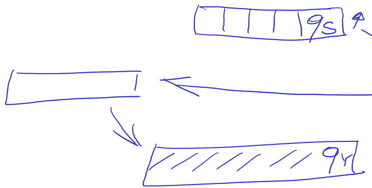
где ϕ — формула в КНФ, $q_i = \forall$ или $q_i = \exists$.

Теорема

QBF **PSPACE**-полна.

Сводим $L \in$ **PSPACE** к QBF.

Достижимость в графе $2^{p(n)}$ конфигураций машины, принимающей L.



$2^{p(n)}$ вершин

длина $< 2^{p(n)}$

PSPACE-полная задача: `QBF`

Язык `QBF` состоит из замкнутых истинных формул вида

$$\mathrm{q}_1 x_1 \mathrm{q}_2 x_2 \ldots \phi,$$

где ϕ — формула в КНФ, $\mathrm{q}_i = \forall$ или $\mathrm{q}_i = \exists$.

Теорема

`QBF` **PSPACE**-полна.

Сводим $L \in$ **PSPACE** к `QBF`.

Достижимость в графе $2^{p(n)}$ конфигураций машины, принимающей L.

Строим $\phi_i(c_1, c_2) =$ «существует путь из $c_1 \rightsquigarrow c_2$ длины $\leq 2^i$».
$\phi_i(c_1, c_2) = \exists d\ \phi_{i-1}(c_1, d) \wedge \phi_{i-1}(d, c_2)$.
? $\phi_{p(n)}(\text{стартовая}, \text{принимающая})$

$$c_1 \xrightarrow[\le 2^{i-1}]{} d \xrightarrow[\le 2^{i-1}]{} c_2$$

PSPACE-полная задача: QBF

Язык QBF состоит из замкнутых истинных формул вида

$\Phi' \rightsquigarrow \Phi$ в КНФ

$$q_1 x_1 q_2 x_2 \dots \phi,$$

схема

$\exists$доп. пер.

где ϕ — формула в КНФ, $q_i = \forall$ или $q_i = \exists$.

Теорема

QBF **PSPACE**-полна.

$\Phi_{p(n)}$(ст, прин.)

$x \in L \longrightarrow F \in QBF$

Сводим $L \in$ **PSPACE** к QBF.

Достижимость в графе $2^{p(n)}$ ($|x|$) конфигураций машины, принимающей L.

Строим $\phi_i(c_1, c_2) =$ «существует путь из $c_1 \rightsquigarrow c_2$ длины $\leq 2^i$».

~~$\phi_i(c_1, c_2) = \exists d\, \phi_{i-1}(c_1, d) \wedge \phi_{i-1}(d, c_2)$.~~

кванторы

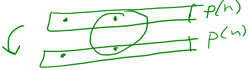
$\phi_i(c_1, c_2) = \exists d \forall x \forall y (((x = c_1 \wedge y = d) \vee (x = d \wedge y = c_2)) \Rightarrow \phi_{i-1}(x, y))$.

($\phi_0(c_1, c_2)$ записывается как в теореме Кука-Левина.)

? $\phi_{p(n)}$(стартовая, принимающая)

$c_1 \mapsto c_2$

$3 \times p(n)$ $\exists d_0 \exists d_1 \exists d_2 \dots$



PSPACE-полная задача: QBF

Язык QBF состоит из замкнутых истинных формул вида

$$\mathrm{q}_1 x_1 \mathrm{q}_2 x_2 \ldots \phi,$$

где ϕ — формула в КНФ, $\mathrm{q}_i = \forall$ или $\mathrm{q}_i = \exists$.

Теорема

QBF **PSPACE**-полна.

Сводим $L \in$ **PSPACE** к QBF.

Достижимость в графе $2^{p(n)}$ конфигураций машины, принимающей L.

Строим $\phi_i(c_1, c_2) =$ «существует путь из $c_1 \rightsquigarrow c_2$ длины $\leq 2^i$».
$\cancel{\phi_i(c_1, c_2) = \exists d\ \phi_{i-1}(c_1, d) \wedge \phi_{i-1}(d, c_2).}$
$\phi_i(c_1, c_2) = \exists d \forall x \forall y (((x = c_1 \wedge y = d) \vee (x = d \wedge y = c_2)) \Rightarrow \phi_{i-1}(x, y)).$
($\phi_0(c_1, c_2)$ записывается как в теореме Кука-Левина.)
? $\phi_{p(n)}(\text{стартовая}, \text{принимающая})$

Следствие

PH = **PSPACE** $\Rightarrow$

PSPACE-полная задача: QBF

Язык QBF состоит из замкнутых истинных формул вида

$$\mathrm{q}_1 x_1 \mathrm{q}_2 x_2 \ldots \phi,$$

где ϕ — формула в КНФ, $\mathrm{q}_i = \forall$ или $\mathrm{q}_i = \exists$.

Теорема

QBF **PSPACE**-полна.

Сводим $L \in$ **PSPACE** к QBF.

Достижимость в графе $2^{p(n)}$ конфигураций машины, принимающей L.

Строим $\phi_i(c_1, c_2) =$ «существует путь из $c_1 \rightsquigarrow c_2$ длины $\leq 2^i$».
$\cancel{\phi_i(c_1, c_2) = \exists d\ \phi_{i-1}(c_1, d) \wedge \phi_{i-1}(d, c_2).}$
$\phi_i(c_1, c_2) = \exists d \forall x \forall y (((x = c_1 \wedge y = d) \vee (x = d \wedge y = c_2)) \Rightarrow \phi_{i-1}(x, y)).$
($\phi_0(c_1, c_2)$ записывается как в теореме Кука-Левина.)
? $\phi_{p(n)}(\text{стартовая}, \text{принимающая})$

Следствие

PH = **PSPACE** $\Rightarrow$ **PH** коллапсирует

Иерархия по памяти

Теорема

$\mathbf{DSpace}[s(n)] \neq \mathbf{DSpace}[S(n)]$, где $s(n) = o(S(n))$ и $\forall n > n_0\ S(n) \geq \log n$.

$$L = \left\{ x = \underline{M01^k} \,\middle|\, \begin{array}{l} k \in \mathbb{N} \cup \{0\}, \\ |M| < S(|x|), \\ \underline{M \text{ отвергает } x \text{ с памятью} \leq S(|x|)} \end{array} \right\} \in \mathbf{DSpace}[S(|x|)].$$

От противного:

Пусть M_* распознаёт L с памятью $s_*(|x|) = O(s(|x|))$.

$\underline{\exists N_1}\ \forall n > N_1\ \ \underline{s_*(n) < S(n)}$.

Рассмотрим $N_* > \max\{N_1, \underline{2^{|M_*|}}\}$.

Если $M_*(\underbrace{M_*01^{N_* - |M_*| - 1}}_{x}) = 1 \Rightarrow$ её аргумент $\notin L$, и наоборот.

$= 0$ $\qquad x \in L$

Если памяти совсем мало...

Теорема

$\mathbf{DSpace}[\log\log n] \neq \mathbf{DSpace}[O(1)]$.

Теорема

$\forall\,\varepsilon > 0\ \ \mathbf{DSpace}[(\log\log n)^{1-\varepsilon}] = \mathbf{DSpace}[O(1)]$.

Иерархия по времени

$$\underbrace{P \subseteq NP \subseteq \ldots \subseteq PSPACE \subseteq EXP}_{\neq} = \bigcup_k DTime[2^{n^k}]$$

Теорема

$\mathbf{DTime}[t(n)] \neq \mathbf{DTime}[T(n)]$, где $\underline{t(n)\log t(n) = o(T(n)),}$ $T(n) = \Omega(n)$.

$$L = \left\{ x = M01^k \;\middle|\; \begin{array}{l} k \in \mathbb{N} \cup \{0\}, \\ |M| < T(|x|), \\ M \text{ отвергает } x \text{ за время} \leq \dfrac{T(|x|)}{\log T(|x|)} \end{array} \right\} \in \mathbf{DTime}[T(|x|)].$$

$$\cdot \log \frac{T}{\log T}$$

Здесь используется эффективная универсальная МТ, моделирующая $f(n)$ шагов произвольной машины (с произвольным числом лент) за $O(f(n)\log f(n))$ шагов. Остальное аналогично иерархии по памяти.

Полиномиальные схемы

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

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

Полиномиальные схемы

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

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

Полиномиальные схемы:

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

Ясно, что $\mathbf{P} \subsetneq \mathbf{P/poly}$.
нераз.

$L = \{ 1^n \mid n$ – остан. МТ $n(n) = 1 \}$

(или другой неразр.)

$$C_{\underline{n}}(x) = \begin{cases} 0, & 1^n \notin L \\ 1, & 1^n \in L \\ 0, & x \neq 1^n \end{cases}$$

Полиномиальные схемы

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

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

non-uniform

Полиномиальные схемы:

uniform

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

Ясно, что $\mathbf{P} \subsetneq \mathbf{P/poly}$.

подсказки
advice

Альтернативное определение:

...если имеются $R \in \mathbf{P}$ и последовательность строк $\{y_n\}_{n \in \mathbb{N}}$ полин. длины, т.ч.

$$\forall x \; (x \in L \Leftrightarrow R(x, y_{|x|}) = 1).$$

$\Rightarrow \; \{C_n\}$ — подсказки , $R(x, C_{|x|}) = C_{|x|}(x)$

$\Leftarrow \; R \rightsquigarrow C_n = C'_n(\cdot, y_n)$

Теорема Карпа-Липтона

Теорема

$\mathbf{NP} \subseteq \mathbf{P}/\mathbf{poly} \Rightarrow \mathbf{PH} = \Sigma^2\mathbf{P}$.

Покажем, что $\Sigma^3\mathbf{P}$-полный язык

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

лежит в $\Sigma^2\mathbf{P}$.

Теорема Карпа-Липтона

Теорема

$\mathbf{NP} \subseteq \mathbf{P/poly} \Rightarrow \mathbf{PH} = \Sigma^2\mathbf{P}$.

Покажем, что $\Sigma^3\mathbf{P}$-полный язык

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

лежит в $\Sigma^2\mathbf{P}$.

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

$\mathbf{NP} \subseteq \mathbf{P/poly} \Rightarrow \mathbf{PH} = \Sigma^2\mathbf{P}$.

Покажем, что $\Sigma^3\mathbf{P}$-полный язык

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

лежит в $\Sigma^2\mathbf{P}$.

$(\exists x \forall y \exists z\ F) \in \mathtt{QBF}_3 \Leftrightarrow$
$\exists$ 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 $f\colon \{0,1\}^n \to \{0,1\}$, зависящая от первых $c \cdot k \cdot \log n$ битов, для которой нет булевой схемы размера n^k.

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

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

Схемы фиксированного полиномиального размера

Теорема

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

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

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

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

Схемы фиксированного полиномиального размера

Теорема

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

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

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

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

Схемы фиксированного полиномиального размера

Теорема

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

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

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

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

Остаётся превратить n^k в $O(n^k)$.

Схемы фиксированного полиномиального размера

Теорема

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

Следствие

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

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

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 **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 **P**.

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

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

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

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