Структурная теория сложности

Эдуард Алексеевич Гирш

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

ПОМИ РАН

12 октября 2008 г.

Напоминание: оракулы

Оракульная МТ имеет доступ к оракулу, который за 1 шаг даёт ей ответ на вопрос.

Формально: состояния q_{in}, q_{out} и "фантастический переход" из q_{in} в q_{out}, заменяющий содержимое [третьей] ленты на ответ оракула.

M^B — оракульная машина M, которой дали конкретный оракул B.

Напоминание: оракулы

Оракульная МТ имеет доступ к оракулу, который за 1 шаг даёт ей ответ на вопрос.

Формально: состояния q_{in}, q_{out} и "фантастический переход" из q_{in} в q_{out}, заменяющий содержимое [третьей] ленты на ответ оракула.

M^B — оракульная машина M, которой дали конкретный оракул B.

Для классов $\mathcal{C}$, $\mathcal{D}$ новый класс

$$\mathcal{C}^{\mathcal{D}}$$

состоит из языков вида C^D, где $D \in \mathcal{D}$, C — *машина* для языка из $\mathcal{C}$.

Классы дополнений

$$\textbf{co-}\mathcal{C} = \{L \mid \overline{L} \in \mathcal{C}\}.$$

Например,

$$\texttt{SAT} \in \textbf{NP},$$

а

$$\{\text{всюду ложных формул}\} \in \textbf{co-NP}.$$

Полиномиальная иерархия

$$\Sigma^0\mathbf{P} = \Pi^0\mathbf{P} = \Delta^0\mathbf{P} = \mathbf{P},$$

$$\begin{aligned}
\Sigma^{i+1}\mathbf{P} &= \mathbf{NP}^{\Pi^i\mathbf{P}}, \\
\Pi^{i+1}\mathbf{P} &= \mathbf{co\text{-}NP}^{\Sigma^i\mathbf{P}}, \\
\Delta^{i+1}\mathbf{P} &= \mathbf{P}^{\Sigma^i\mathbf{P}}.
\end{aligned}$$

$$\mathbf{PH} = \bigcup_{i \geq 0} \Sigma^i\mathbf{P}.$$

Полиномиальная иерархия

$$\Sigma^0\mathbf{P} = \Pi^0\mathbf{P} = \Delta^0\mathbf{P} = \mathbf{P},$$

$$\begin{aligned}
\Sigma^{i+1}\mathbf{P} &= \mathbf{NP}^{\Sigma^i\mathbf{P}}, \\
\Pi^{i+1}\mathbf{P} &= \mathbf{co\text{-}NP}^{\Sigma^i\mathbf{P}}, \\
\Delta^{i+1}\mathbf{P} &= \mathbf{P}^{\Sigma^i\mathbf{P}}.
\end{aligned}$$

$$\mathbf{PH} = \bigcup_{i \geq 0} \Sigma^i\mathbf{P}.$$

Классы полиномиальной иерархии через кванторы

Теорема

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное отношение $R \in \Pi^{k-1}\mathbf{P}$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y R(x, y))$.

Классы полиномиальной иерархии через кванторы

Теорема

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное отношение $R \in \Pi^{k-1}\mathbf{P}$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y R(x, y))$.

Следствие

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное $R \in P$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots R(x, y_1, y_2, \ldots, y_k))$.

$L \in \Pi^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное $R \in P$, такое, что $\forall x\ (x \in L \Leftrightarrow \forall y_1 \exists y_2 \forall y_3 \ldots R(x, y_1, y_2, \ldots, y_k))$.

Классы полиномиальной иерархии через кванторы

Теорема

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное отношение $R \in \Pi^{k-1}\mathbf{P}$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y R(x, y))$.

$\boxed{\Leftarrow}$ L распознаётся следующей машиной с оракулом R: недетерминированно выберем y и проверим $R(x, y)$.

Классы полиномиальной иерархии через кванторы

Теорема

$L \in \Sigma^k\mathbf{P} \Leftrightarrow \exists$ полиномиально ограниченное отношение $R \in \Pi^{k-1}\mathbf{P}$, такое, что $\forall x\ (x \in L \Leftrightarrow \exists y R(x, y))$.

$\boxed{\Rightarrow}$ Индукция. База: определение $\Sigma^1\mathbf{P} = \mathbf{NP}$.

Переход $(k-1 \to k)$:
$L = L(M^O)$, где M — полин. НМТ, $O \in \Sigma^{k-1}\mathbf{P}$, по предп. индукции имеется п.о. $S \in \Pi^{k-2}\mathbf{P}$, т.ч. $\forall q(q \in O \Leftrightarrow \exists w\, S(q, w))$.

Строим R:
$R(x, y) = 1$, если y — принимающая ветвь вычисления M^O, но Y-ответы оракула снабжены сертификатами w: $S(q, w) = 1$.

$R \in \Pi^{k-1}\mathbf{P}$: детерминированно проверяем корректность y, затем комбинируем $\Pi^{k-1}\mathbf{P}$-вычисления, проверяющие N-ответы, и $\Pi^{k-2}\mathbf{P}$-вычисления, проверяющие сертификаты Y-ответов [см. доску].

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

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

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

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

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

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

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

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

Классы, ограниченные по времени и памяти

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

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

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

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

$f(n)$ 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

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

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

Теорема

QBF **PSPACE**-полна.

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.

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.

Построим граф $2^{p(n)}$ конфигураций машины, принимающей L. Решим задачу достижимости.

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

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

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

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.

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.

Следствие

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

Иерархия по памяти

Теорема

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

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

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

$\exists N_1\ \forall n > N_1 \quad s_*(n) < S(n)$.

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

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

Если памяти совсем мало...

Теорема

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

Теорема

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

Иерархия по времени

Теорема

$\mathbf{DTime}[t(n)] \neq \mathbf{DTime}[T(n)]$, где

$$\frac{t(n)\zeta(T(n))\log T(n)}{T(n)} \to 0,$$

$T(n) = \Omega(n)$ и ζ — легко вычислимая сколь угодно медленно растущая функция (например, log или log log).

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

Здесь универсальная МТ должна промоделировать $f(n)$ шагов произвольной машины (с произвольным числом лент k) за $kf(n)\log f(n)$ шагов — см. на доску.

Иерархия по времени для НМТ

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

Теорема

$t(n+1) = o(T(n)) \Rightarrow \mathbf{NTime}[t(n)] \neq \mathbf{NTime}[T(n)]$.

- x принимается — значит, $\exists$ принимающее вычисление;

Иерархия по времени для НМТ

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

Теорема

$t(n+1) = o(T(n)) \Rightarrow \mathbf{NTime}[t(n)] \neq \mathbf{NTime}[T(n)]$.

- x принимается — значит, $\exists$ принимающее вычисление;
- просто сказать противоположное в конце не даст гарантировано отвергнуть x;

Иерархия по времени для НМТ

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

Теорема

$t(n+1) = o(T(n)) \Rightarrow \mathbf{NTime}[t(n)] \neq \mathbf{NTime}[T(n)].$

- x принимается — значит, $\exists$ принимающее вычисление;
- просто сказать противоположное в конце не даст гарантировано отвергнуть x;
- сопоставим каждой НМТ M_i длину входа n_i т.ч. $n_{i+1} > 2^{n_i}$;

Иерархия по времени для НМТ

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

Теорема

$t(n+1) = o(T(n)) \Rightarrow \mathbf{NTime}[t(n)] \neq \mathbf{NTime}[T(n)].$

- x принимается — значит, $\exists$ принимающее вычисление;
- просто сказать противоположное в конце не даст гарантировано отвергнуть x;
- сопоставим каждой НМТ M_i длину входа n_i т.ч. $n_{i+1} > 2^{n_i}$;
- для длины входа $n \in [n_i \,..\, 2^{n_i} - 1]$ пусть

$$M(x) = M_i(1x), \text{ если хватит времени,}$$

т.е. если есть слишком быстрая M_{i_*}, т.ч. $L(M) = L(M_{i_*})$, то

$$M_{i_*}(1^{n_{i_*}}) = M_{i_*}(1^{n_{i_*}+1}) = \ldots = M_{i_*}(1^{2^{n_{i_*}}}).$$

Иерархия по времени для НМТ

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

Теорема

$t(n+1) = o(T(n)) \Rightarrow \mathbf{NTime}[t(n)] \neq \mathbf{NTime}[T(n)].$

- x принимается — значит, $\exists$ принимающее вычисление;
- просто сказать противоположное в конце не даст гарантировано отвергнуть x;
- сопоставим каждой НМТ M_i длину входа n_i т.ч. $n_{i+1} > 2^{n_i}$;
- для длины входа $n \in [n_i \,..\, 2^{n_i} - 1]$ пусть

$$M(x) = M_i(1x), \text{ если хватит времени,}$$

т.е. если есть слишком быстрая M_{i_*}, т.ч. $L(M) = L(M_{i_*})$, то

$$M_{i_*}(1^{n_{i_*}}) = M_{i_*}(1^{n_{i_*}+1}) = \ldots = M_{i_*}(1^{2^{n_{i_*}}}).$$

- на длине 2^{n_i} детерминированно (время есть!) обратим $M_i(1^{n_i})$:

$$M_{i_*}(1^{2^{n_{i_*}}}) = \neg M_{i_*}(1^{n_{i_*}}).$$

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.

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

Полиномиальные схемы:

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

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

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:

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

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

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

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

Эквивалентность — см. на доску.