Введение в теорию сложности вычислений

Эдуард Алексеевич Гирш

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

СПбГУ и ПОМИ РАН

лекция 19 ноября 2020 г.

NP-полная задача: 3-SAT

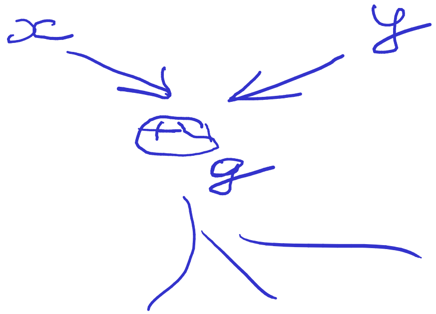
“Теорема Кука–Левина”

$$\widetilde{\text{3-SAT}} = \{(F, A) \mid F \text{ — в 3-КНФ},\ F(A) = 1\}.$$

Пример:

$$(\underbrace{(\neg x \vee \neg y \vee \neg z)} \wedge (y \vee \neg z) \wedge (z),\ [x = 0,\ y = 1,\ z = 1]) \in \widetilde{\text{3-SAT}}.$$

- по переменной для каждого гейта;
- гейт $g(x, y) \mapsto$ клозы, выражающие $g = g(x, y)$, например ($g = \oplus$):



$$\begin{array}{lllll}
(x & \vee & y & \vee & \neg g) \wedge \\
(\neg x & \vee & \neg y & \vee & \neg g) \wedge \\
(x & \vee & \neg y & \vee & g) \wedge \\
(\neg x & \vee & y & \vee & g)
\end{array}$$

output

Clauses g_1 ∧

Clauses g_2 ∧

...

output

- для последнего гейта g клоз (g).

$\widetilde{\mathbf{NP}}$ vs NP

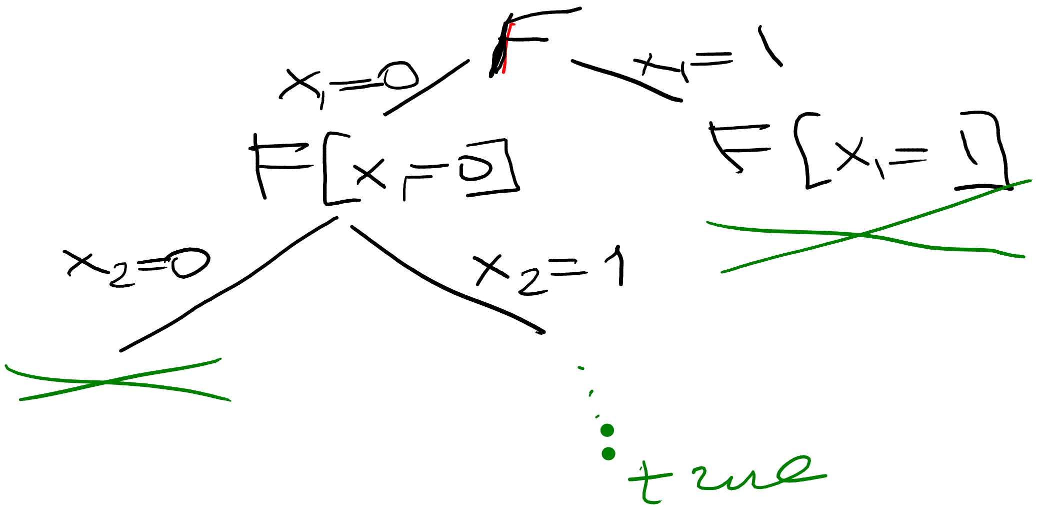
Теорема

$R \in \widetilde{\mathbf{NP}}$, $L(R)$ — **NP**-полон $\Rightarrow R \to \cancel{R(L)}.$ $L(R)$

$R \to \widetilde{\mathrm{SAT}} \to \mathrm{SAT} \to \cancel{R(L)}.$ $L(R)$

Какое значение переменной x_1 в выполняющем наборе для F?
Спросим про $F[x_1 = 0] \in \mathrm{SAT}$ и $F[x_1 = 1] \in \mathrm{SAT}$.

Перейдём к нужной формуле $F[x_1 = v]$ и спросим про следующую переменную.



$\widetilde{\mathbf{NP}}$ vs NP

Теорема

$R \in \widetilde{\mathbf{NP}}$, $R(L)$ — **NP**-полон $\Rightarrow R \to R(L)$.

$R \to \widetilde{\mathrm{SAT}} \to \mathrm{SAT} \to R(L)$.

Какое значение переменной x_1 в выполняющем наборе для F?
Спросим про $F[x_1 = 0] \in \mathrm{SAT}$ и $F[x_1 = 1] \in \mathrm{SAT}$.

Перейдём к нужной формуле $F[x_1 = v]$ и спросим про следующую переменную.

Замечание

"Контрпример" без **NP**-полноты: $\widetilde{\mathrm{FACTOR}}$.

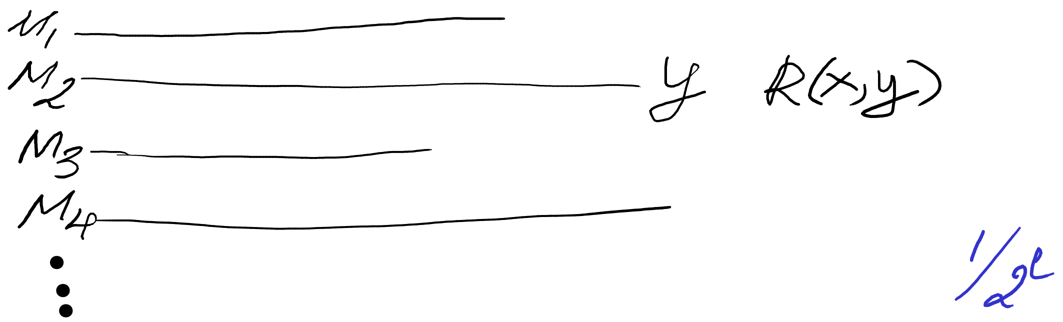
x – составное?

Primality in P

Оптимальный алгоритм для $\widetilde{\mathbf{NP}}$-задачи (Л.Левин)

Хотим решить задачу, заданную отношением R, на входе x.

- Перебираем все машины (не только полиномиальные).
- На этапе номер $\underline{2^{\ell}(1+2k)}$ моделируем k-ый шаг машины M_{ℓ}.
- Если выдан ответ w, проверяем его $R(x, w)$.



Сравним со временем работы конкретной машины M_i:

- Могло бы быть $t(x) \le \mathrm{const}_i \cdot t_i(x) + p(|x|)$ (на проверку), если б можно было моделировать шаг-в-шаг;
- На ДМТ шаг будет стоить дорого, т.к. машин много одновременно, поэтому $t(x) \le \mathrm{const}_i \cdot p(t_i(x))$.

Оптимальный алгоритм для $\widetilde{\mathbf{NP}}$-задачи (Л.Левин)

Хотим решить задачу, заданную отношением R, на входе x.

- Перебираем все машины (не только полиномиальные).
- На этапе номер $2^\ell(1+2k)$ моделируем k-ый шаг машины M_ℓ.
- Если выдан ответ w, проверяем его $R(x,w)$.

Сравним со временем работы конкретной машины M_i:

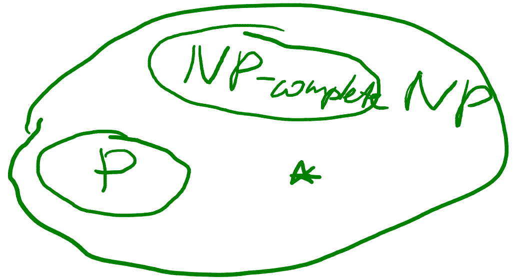
- Могло бы быть $t(x) \le \mathrm{const}_i \cdot t_i(x) + p(|x|)$ (на проверку), если б можно было моделировать шаг-в-шаг;
- На ДМТ шаг будет стоить дорого, т.к. машин много одновременно, поэтому $t(x) \le \mathrm{const}_i \cdot \underline{p(t_i(x))}$.

Если $\mathbf{P} = \mathbf{NP}$, такой алгоритм быстро решает SAT.

Не **NP**-полные задачи в $\mathbf{NP} \setminus \mathbf{P}$

Теорема

Если $\mathbf{P} \neq \mathbf{NP}$, то существует язык $L \in \mathbf{NP} \setminus \mathbf{P}$, <u>не</u> являющийся **NP**-полным.



Не **NP**-полные задачи в $\mathbf{NP} \setminus \mathbf{P}$

$M_1, M_2, \ldots$ — все полиномиальные ДМТ с “будильниками”,
$R_1, R_2, \ldots$ — все полиномиальные сведения с “будильниками”.

$$\mathcal{K} = \{x \mid x \in \mathrm{SAT} \;\wedge\; \underline{f(|x|) \vdots 2}\ \}. \in NP$$

$f(n)$:

(I) за n шагов: вычисляем $\underbrace{f(0), f(1), \ldots, f(i)}_{n} =: \underline{k}$ — сколько успеем;

(II) за n шагов:

$|z| = n'$, $f(n')$

(a) if $k \vdots 2$, if $\exists z$: $M_{k/2}(z) \neq \mathcal{K}(z)$, return $k+1$
(если не успели, return k);

$\mathcal{K} \overset{?}{\in} P$

(b) if $k \not\vdots 2$, if $\exists z$: $\mathcal{K}(R_{(k-1)/2}(z)) \neq \mathrm{SAT}(z)$, return $k+1$;
(если не успели, return k);

SAT $\to$ $\mathcal{K}$
NP-тр.

$\mathcal{K} \in \mathbf{P} \Rightarrow$ навсегда (IIa) $\Rightarrow$ f$\vdots$2 п.в. $\Rightarrow \mathcal{K} \approx \mathrm{SAT}$. $\Rightarrow P = NP$

$\mathcal{K}$ **NP**-полон $\Rightarrow$ навсегда (IIb) $\Rightarrow f \not\vdots 2$ п.в. $\Rightarrow |\mathcal{K}| < \infty$. $\Rightarrow \mathcal{K} \in P \Rightarrow P = NP$

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.

$IP = PSPACE$

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

$IP^A \neq PSPACE^A$

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

NP^A

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

Классы дополнений

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

Например,

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

а

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

Полиномиальная иерархия

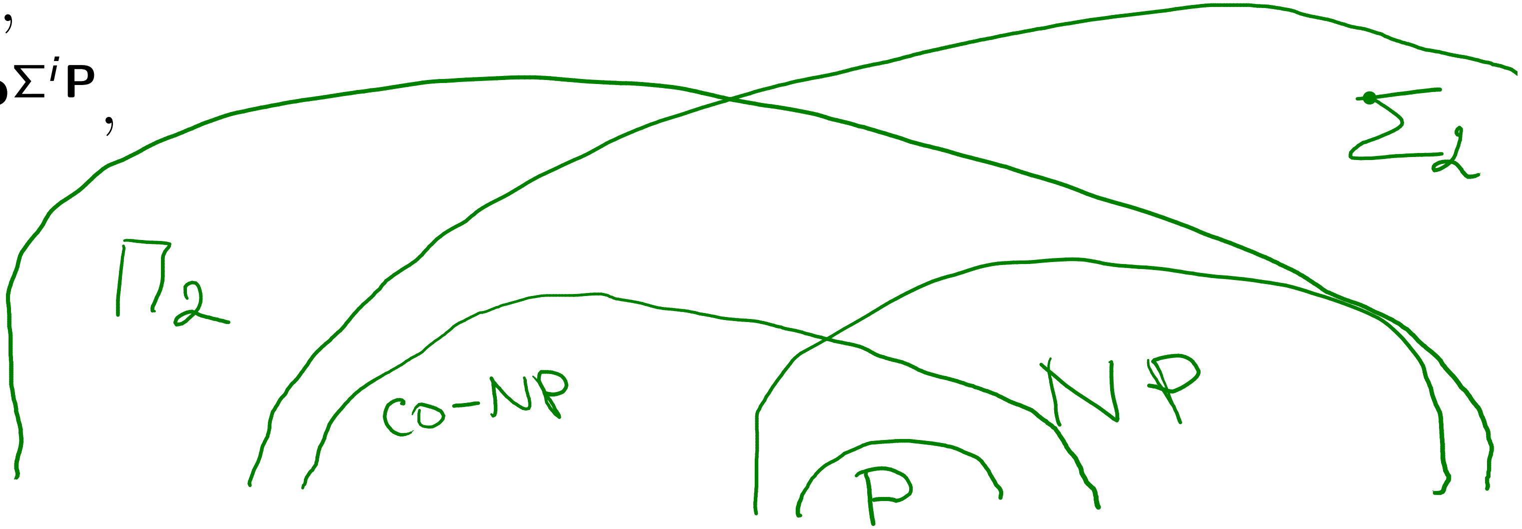
языки

M^A $\qquad$ $M^{\overline{A}}$

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

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

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



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 $\forall x\ (x \in L \Leftrightarrow \exists y R(x, y))$.

$\Sigma^{i+1}\mathbf{P} = \mathbf{NP}^{\Pi^i\mathbf{P}}$
$\Pi^{i+1}\mathbf{P} = \mathbf{co\text{-}NP}^{\Sigma^i\mathbf{P}}$

Классы полиномиальной иерархии через кванторы

Теорема

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

$\Sigma^{i+1}\mathbf{P} = \mathbf{NP}^{\Pi^i\mathbf{P}}$
$\Pi^{i+1}\mathbf{P} = \mathbf{co\text{-}NP}^{\Sigma^i\mathbf{P}}$

Следствие

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

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

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 $R(x,y)$.

$\Sigma^{i+1}\mathbf{P} = \mathbf{NP}^{\Pi^i\mathbf{P}}$
$\Pi^{i+1}\mathbf{P} = \mathbf{co\text{-}NP}^{\Sigma^i\mathbf{P}}$

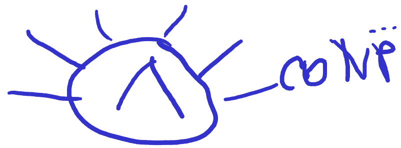
Классы полиномиальной иерархии через кванторы

$F \in SAT$ $G \in SAT$

Теорема

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

$\Sigma^{i+1}\mathbf{P} = \mathbf{NP}^{\Pi^i\mathbf{P}}$
$\Pi^{i+1}\mathbf{P} = \mathbf{co\text{-}NP}^{\Sigma^i\mathbf{P}}$



$\boxed{\Rightarrow}$ 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. $\forall q(q \in O \Leftrightarrow \exists w\, S(q,w))$.

$O(q) = 1 \Leftrightarrow$
$\Leftrightarrow \exists w \quad S(q,w) = 1$

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

$O(r) = 0$
$\overline{O}(r) = 1$
$\overline{O} \in \Pi^{k-1}$

$R \in \Pi^{k-1}\mathbf{P}$: детерминированно проверяем корректность y,
затем комбинируем $\Pi^{k-1}\mathbf{P}$-вычисления, проверяющие No-ответы, и $\Pi^{k-2}\mathbf{P}$-вычисления, проверяющие сертификаты Yes-ответов.