

Московский физико-технический институт  
Физтех-школа прикладной математики и информатики

Б О Л Ъ Ш О Е Н А З ВАНИЕ К У Р С А  
V СЕМЕСТР

Лектор: *Иван Иванович Иванов*

h\nu

Автор: *Павел Дуров*  
*Репозиторий на Github*

осень 2022

## Содержание

<b>1 Класс P</b>	<b>2</b>
1.1 Базовые определения . . . . .	2
1.2 Неконструктивные оценки P . . . . .	2
1.3 Другие классы задач . . . . .	3
1.4 Асимптотики различных задач . . . . .	3
<b>2 Класс NP</b>	<b>3</b>
2.1 Определение через сертификат . . . . .	4
2.2 Некоторые следствия из определений . . . . .	5

# 1 Класс P

## 1.1 Базовые определения

Будем рассматривать задачи на распознавание, т.е. дан  $A \subset \{0, 1\}^*$  и требуется по  $x \in \{0, 1\}^* \mapsto \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$ . Пусть  $M$  решает данную задачу, т.е.  $\forall x(x \in A \Leftrightarrow M(x) = 1)$ .

**Определение 1.1.**  $time_M(x)$  — число шагов  $M(x)$  при вычислении ответа.

**Определение 1.2.**  $time_M(n) = \max_{x:|x|=n} time_M(x)$

**Определение 1.3.**  $time_M(n) = O(f(n))$ , если  $\exists C : \forall n : time_M(n) \leq C \cdot f(n)$

Возникает вопрос: можно ли сказать, что  $time_A(n) = \min_{M: M \text{ решает } A} time_M(n)$ ? Нет, но показать это достаточно сложно (Теорема Блюма).

Поэтому мы приходим к данному определению:

**Определение 1.4.**  $\mathbf{DTIME}(t(n)) = \{A | \exists M : M(x) = 1 \Leftrightarrow x \in A, time_M(n) = O(t(n))\}$

Заметим, что для определения **DTIME**, необходимо задать модель вычислений. Обычно такой моделью выбирают многоленточную машину Тьюринга.

**Определение 1.5.**  $P = \mathbf{DTIME}(poly(n)) = \bigcup_{k=1}^{\infty} \mathbf{DTIME}(n^k)$

**Тезис Черча-Тьюринга в сильной форме:** Любая задача, эффективно решаемая физическим устройством, решается за полиномиальное время на машине Тьюринга.

**Пример** (Нетривиальные примеры задач из **P**). 1.  $\mathbb{P}$  — множество простых чисел

2. Линейное программирование — как пример, нахождения максимума функции на многограннике. Эта задача не бинарная, но вот задача "достижима ли это число на многограннике" принадлежит классу **P**.
3. Симплекс-метод — алгоритм решения

## 1.2 Неконструктивные оценки P

Рассмотрим, например, задачу определения графа на планарность. Для этого существует два критерия: критерий Понtryгина-Куратовского: граф планарен  $\Leftrightarrow$  в нем нет подграфов, гомеоморфных  $K_5, K_{3,3}$ . Также, существует критерий Вагнера: граф планарен  $\Leftrightarrow$  в нем нет миноров  $K_5, K_{3,3}$  (минор — граф, полученный из исходного удалением и стягиванием ребер). Рассмотрим свойства, которые сохраняются при удалении и стягивании ребер.

**Теорема 1.1** (Робертсона-Сеймура). 1. Для любого свойства, аналогичного планарности выполнен аналог критерия Вагнера с конечным числом запрещенных миноров.

2. Наличие такого минора проверяется за полиномиальное время

**Следствие.** Любое такое свойство лежит в классе  $\mathbb{P}$ .

Но проблема в том, что мы не знаем миноров, которые необходимо проверить, чтобы найти проверить выполнение данного свойства.

### 1.3 Другие классы задач

**Определение 1.6.**  $\text{QP} = \text{DTIME}(2^{\text{poly}(\log(n))}) = \bigcup_{c=1}^{\infty} \text{DTIME}(2^{(\log n)^c})$

**Определение 1.7.**  $\text{E} = \text{DTIME}(2^{O(n)}) = \bigcup_{c=1}^{\infty} \text{DTIME}(2^{cn})$

**Определение 1.8.**  $\text{EXP} = \text{DTIME}(2^{\text{poly}(n)}) = \bigcup_{c=1}^{\infty} \text{DTIME}(2^{n^c})$

**Определение 1.9.**  $\text{EE} = \text{DTIME}(2^{2^{cn}}) = \bigcup_{c=1}^{\infty} \text{DTIME}(2^{2^{cn}})$

### 1.4 Асимптотики различных задач

**Пример.**  $\text{LOG-CLIQUE} = \{G \mid \omega(G) \geq \log_2 n\} \in \text{QP}$ . Является квазиполиномиальной, т.к.  $C_n^{\log n} \leq n^{\log n}$ , т.е. полный перебор осуществляется за квазиполином

**Пример** (Задача о доминирующем множестве в турнире). непон

**Пример.**  $\text{GI} = \{(G_1, G_2) : G_1 \cong G_2\} \in \text{QP}$

**Пример.**  $\text{3COL} = \{G : \chi(G) \leq 3\} \in E$

**Теорема 1.2** (об иерархии по времени). *Если  $f \ll g \Rightarrow \text{DTIME}(f(\mathbf{n})) \subsetneq \text{DTIME}(g(\mathbf{n}))$*

## 2 Класс NP

Сейчас будем рассматривать модель вычислений — *недетерминированную машину Тьюринга* или НМТ. В отличие от обычной машины Тьюринга, функция перехода теперь многозначна (по аналогии с ДКА и НКА).

Соответственно, время работы такой машины Тьюринга —  $\text{time}_M(x) = \max \# \text{шагов по всем вариантам перехода}$ .

**Замечание.** Можем считать, что дерево переходов двоичное. Действительно, размер ветвлений ограничено мощностью  $|\Sigma| \cdot |Q| \cdot |\{N, R, L\}|$  — некоторая константа, не зависящая от входа. Тогда каждое  $m$ -ветвление можно заменить  $\log_2 m$  2-ветвлениями.

Ответ данной машины будем понимать следующее:

$$M(x) = \begin{cases} 1, & \text{существует принимающая ветка} \\ 0, & \text{иначе} \end{cases}$$

**Замечание.** Ответ вычисляется как дизъюнкция по всем результатам работы машины

**Определение 2.1.**  $\text{NTIME}(t(n))$  — класс языков, распознаваемых на НМТ за  $O(t(n))$  шагов

**Определение 2.2.**  $\text{NP} = \bigcup_{c=1}^{\infty} \text{NTIME}(n^c)$

**Определение 2.3.**  $\text{NE} = \bigcup_{c=1}^{\infty} \text{NTIME}(2^{cn})$

**Определение 2.4.**  $\text{NEXP} = \bigcup_{c=1}^{\infty} \text{NTIME}(2^{n^c})$

**Замечание.**  $\text{NTIME}(t(n)) \subset \text{DTIME}(2^{t(n)})$

**Замечание.**  $\text{NP} \subset \text{EXP}$  — за время  $\text{EXP}$  можно построить все дерево и вычислить ответ по определению.

## 2.1 Определение через сертификат

**Теорема 2.1.**  $A \in \text{NP} \Leftrightarrow \exists V(x, s) - \text{ДМТ, т.ч. } x \in A \Leftrightarrow \exists s : V(x, s) = 1 \text{ и } V(x, s)$  работает за  $\text{poly}(|x|)$

*Доказательство.*

- $\Leftarrow$  Рассмотрим следующую НМТ, которая сначала печатает все возможные варианты  $s$  (достаточно написать полиномиальное количество символов, т.к. больше  $V$  не сможет прочесть), а потом на входе  $x, s$  запускает  $V$ . Таким образом, Получили машину  $M$ , которая в какой-то ветке напечатает нужный сертификат  $s$  и выведет  $V(x, s) = 1$ .
- $\Rightarrow$  Возьмем в качестве сертификата код нужной ветви в машине  $M$  (0, если надо идти вправо, 1, если влево). Оно и будет нашим сертификатом  $s$ . Машина  $V$  будет спускаться, в соответствии с сертификатом, по дереву переходов. Тогда сертификат существует  $\Leftrightarrow$  существует принимающая ветвь  $\Leftrightarrow A \in \text{NP}$ .

□

**Упражнение.** Сформулировать и доказать аналогичную теорему для классов **NE**, **NEXP**

**Утверждение 2.1.** 1.  $A, B \in \text{P} \Rightarrow A \cap B, A \cup B, \overline{A} \in \text{P}$

2.  $A, B \in \text{NP} \Rightarrow A \cap B, A \cup B \in \text{NP}$

**Замечание.** Вообще говоря,  $\overline{A} \in \text{NP}$  — открытый вопрос. Нельзя просто инвертировать значение машины  $M$  (пусть мы получим машину  $\overline{M}$ ): тогда  $x \in \overline{A} \Leftrightarrow$  все ветки  $\overline{M}$  принимающие, а это не то, что мы хотим. Таким образом, мы приходим к следующему определению:

**Определение 2.5.**  $\text{coNP} = \{A \in \{0, 1\}^* : \overline{A} \in \text{NP}\}$ .

**Замечание.** Аналогично можно доказать, что  $A \in \text{coNP}$  тогда и только тогда, когда ответ вычисляется как конъюнкция всех результатов работы машины или тогда и только тогда, когда  $\exists V : x \in A \Leftrightarrow \forall s V(x, s) = 1$  и  $V$  вычисляется полиномиально от длины  $x$ .

**Пример.** 1.  $\text{SAT} = \{\varphi : \exists x : \varphi(x) = 1\} \in \text{NP}$ .

Также имеет смысл рассмотреть двойственную задачу (к задаче опровергимости формулы):

2.  $\text{TAUT} = \{\varphi : \exists x : \varphi(x) = 1\} \in \text{coNP}$

**Замечание.** Несмотря на доказанную теорему о полноте, вывод не будет являться сертификатом. Действительно, вывод, вообще говоря, не обязан быть полиномиальным и, в таком случае, машина не сможет полностью его прочесть (т.к. работает полиномиально от  $|x|$ )

Отдельный интерес у людей науки представляет множество  $(\text{coNP} \cap \text{NP}) \setminus \text{P}$ . Рассмотрим следующую задачу:

**Определение 2.6.**  $\text{FACTORING} = \{(n, a, b) : \exists d \in (a, b) : d — \text{простое и } n \mid d\}$

**Утверждение 2.2.**  $\text{FACTORING} \in \text{NP} \cap \text{coNP}$ .

*Доказательство.*

$\in \text{NP}$  — сертификат

$\in \text{coNP}$  сертификат — разложение на простые, каждое из которых  $\notin (a, b)$ .

□

## 2.2 Некоторые следствия из определений

**Утверждение 2.3.**  $\text{P} = \text{NP} \Leftrightarrow \text{P} = \text{coNP}$

**Утверждение 2.4.** Следует из того, что  $\text{coP} = \text{P}$  ( $\text{P}$  замкнут относительно дополнения).

**Замечание.** Тем не менее, может быть, что  $\text{P} \neq \text{NP}$ , но  $\text{NP} = \text{coNP}$

**Определение 2.7.**  $A \leq_p B$  сводится по Карпу (сводится полиномиально), если  $\exists$  всюду полиномиально вычислимая от  $|x|$  функция  $f(x)$ , такая, что  $x \in A \Leftrightarrow f(x) \in B$ .

**Определение 2.8.**  $\text{INDSET} = \{(G, k) : \text{в графе } G \text{ есть антиклика из } k \text{ вершин}\}$

**Пример.**  $\text{CLIQUE} \leq_p \text{INDSET}$ . Действительно,  $f(G, k) = f(\bar{G}, k)$  (дополнение по ребрам).

**Определение 2.9.**  $\text{4COL} = \{G : \exists \text{правильная раскраска в 4 цвета}\}$

**Пример.**  $\text{4COL} \leq_p \text{SAT}$ . Мы так уже делали на матлоге, когда сводили некоторые задачи к задачам выполнимости формулы. Для каждой вершины заведем две переменные  $p_i, q_i$ , отвечающие за цвет. Нам нужно для каждого ребра записать, что две вершины, являющиеся его концами, имеют разный цвет, и взять конъюнкцию, т.е:

$$\bigwedge_{(i,j) \in E} (p_i \neq p_j) \vee (q_i \neq q_j)$$

Размер данной формулы будет полиномиальным относительно размера графа.

**Замечание** (Свойства  $\leq_p$ ).

1.  $A \leq_p B, B \leq_p C \Rightarrow A \leq_p C$
2.  $A \leq_p B, B \in \text{P} \Rightarrow A \in \text{P}$
3.  $A \leq_p B, B \in \text{NP} \Rightarrow A \in \text{NP}$
4.  $A \leq_p B \Rightarrow \bar{A} \leq_p \bar{B}$

**Определение 2.10.** Задача  $B \in \text{NPH}$  ( $\text{NP}$ -трудной), если  $\forall A \in \text{NP} : A \leq_p B$ .

**Определение 2.11.**  $\text{NPC} = \text{NP} \cap \text{NPH}$  ( $\text{NP}$ -полные)

**Следствие.**

1.  $B \in \text{NPH}, B \leq_p C \Rightarrow C \in \text{NPH}$
2.  $B \in \text{NPC}, B \leq_p C, C \in \text{NP} \Rightarrow C \in \text{NPC}$

**Утверждение 2.5.**

1.  $P \cap NPH \neq \emptyset \Rightarrow P = NP$
2.  $\text{coNP} \cap NPH \neq \emptyset \Rightarrow NP = \text{coNP}$

*Доказательство.*

2.  $B \in NPH, \text{coNP} \Rightarrow \bar{B} \in NP$ . Теперь, если  $A \leqslant_p B \Rightarrow \bar{A} \leqslant_p \bar{B}$ . Отсюда получаем, что  $\bar{A} \in NP$  и тогда  $NP \subset \text{coNP}$ . Тогда:

$$S \in \text{coNP} \Rightarrow \bar{S} \in NP \Rightarrow \bar{S} \in \text{coNP} \Rightarrow S \in NP$$

□