

Введение в теорию сложности вычислений

Эдуард Алексеевич Гирш

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

СПбГУ и ПОМИ РАН

лекция 5 ноября 2020 г.

Напоминание: что мы решаем (1)

$\emptyset, 10, 100$

$0 \rightarrow 00$
 $1 \rightarrow 11$
 $2 \rightarrow 01$

Работаем в алфавите $\{0, 1\}$.

Множество слов длины n в нём: $\{0, 1\}^n$.

Множество всех (конечных) слов: $\{0, 1\}^*$.

Длина слова x : $|x|$.

Язык (задача распознавания, decision problem): $L \subseteq \{0, 1\}^*$.


Индивидуальная задача — пара (**условие**, **решение**) $\in \{0, 1\}^* \times \{0, 1\}^*$.

Массовая задача — некоторое множество индивидуальных задач, т.е. бинарное отношение на $\{0, 1\}^*$.

Наиболее интересные массовые задачи — бесконечные, с возможностью проверить корректность решения.

Напоминание: что мы решаем (2)

Пример (полагаем $\mathbb{N} \subset \{0, 1\}^*$):

$$\widetilde{\text{FACTOR}} = \{(n, d) \mid n \dot{:} d, 1 < d < n\}.$$


Алгоритм решает задачу поиска для массовой задачи R , если для условия x он находит решение w , удовлетворяющее $(x, w) \in R$.

Напоминание: что мы решаем (2)

Пример (полагаем $\mathbb{N} \subset \{0, 1\}^*$):

$$\widetilde{\text{FACTOR}} = \{(n, d) \mid n \dot{:} d, 1 < d < n\}.$$

Алгоритм решает задачу поиска для массовой задачи R , если для условия x он находит решение w , удовлетворяющее $(x, w) \in R$.

Массовой задаче, заданной отношением R , соответствует язык

$$L(R) = \{x \mid \exists w (x, w) \in R\}.$$

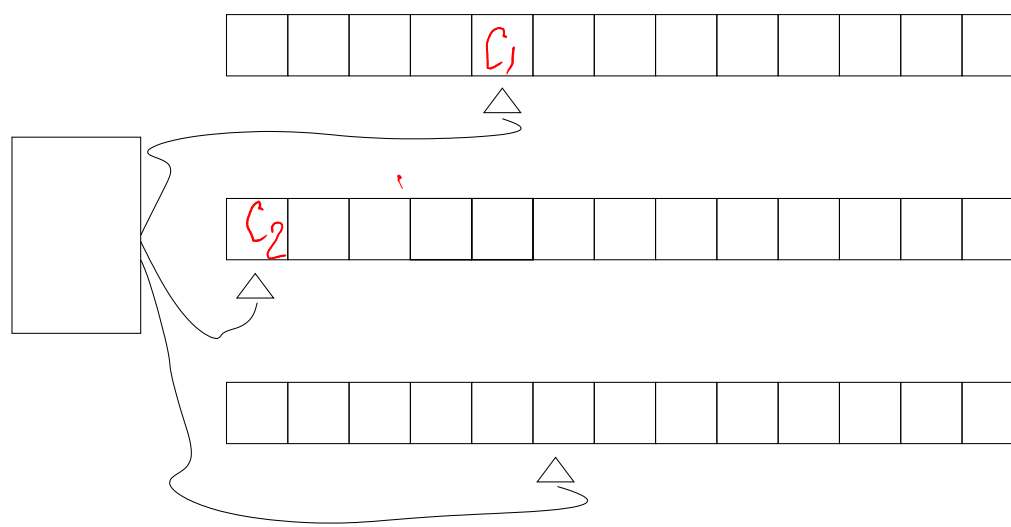
Например,

$$L(\widetilde{\text{FACTOR}}) = \text{множество всех составных чисел.}$$

Напоминание: чем мы решаем (ДМТ — определение)

Детерминированная машина Тьюринга (ДМТ):

- ▶ конечный **алфавит** (с началом ленты и пробелом): $\Sigma = \{0, 1, \triangleright, _ \}$;
- ▶ несколько **лент**, т.е. массивов, бесконечных в одну сторону;
- ▶ читающие/пишущие **головки**, по одной для каждой ленты, каждая видит в один момент одну позицию;
- ▶ конечное множество **состояний**, в т.ч. **начальное** q_S , **принимающее** q_Y и **отвергающее** q_N ;
- ▶ **управляющее устройство (программу)**, содержащее для каждого q, c_1, \dots, c_k одну инструкцию вида $(q, c_1, \dots) \mapsto (q', c'_1, \dots, d_1, \dots)$, где $q, q' \in Q$; $c_i, c'_i \in \Sigma$ — символы, обозреваемые головками; $d_i \in \{\leftarrow, \rightarrow, \cdot\}$ — направление движения.



input read-only

k working tapes

output write-only

Напоминание: чем мы решаем (ДМТ — вычисление)

Вычисление на ДМТ:

- ▶ начало работы:
 - ▶ состояние q_S ;
 - ▶ на первой ленте **вход** (входное слово) и пробелы, остальные ленты заполнены пробелами;
 - ▶ головки в крайней левой позиции;
- ▶ шаг за шагом выполняются инструкции программы;
- ▶ конец работы: когда машина попадает в состояние q_Y либо q_N .

ДМТ **принимает** входное слово, если она заканчивает свою работу в q_Y .

ДМТ **отвергает** q_N .

ДМТ M **распознаёт язык** A , если принимает все $x \in A$, отвергает все $x \notin A$.

Пишем $A = L(M)$.

ДМТ может также **вычислять функцию** (решать задачу поиска).

Значением этой функции на данном входе будем считать содержимое выходной ленты после достижения q_Y .

Напоминание: чем мы решаем (ДМТ — сложность)

Время работы машины M на входе x — количество шагов (применений инструкций) до достижения q_Y или q_N .

Используемая **память** — суммарное крайнее правое положение всех головок на рабочих лентах.

При сублинейных ограничениях на память будет важно, что

- ▶ входная лента **read-only**,
- ▶ выходная лента **write-only**
- ▶ и положения головки на них не считаются.

Двух лент достаточно

Теорема

Для любого $k \in \mathbb{N}$, работу ДМТ M с k рабочими лентами, работающую t шагов, можно промоделировать на ДМТ с двумя рабочими лентами за время $O(t \log t)$.

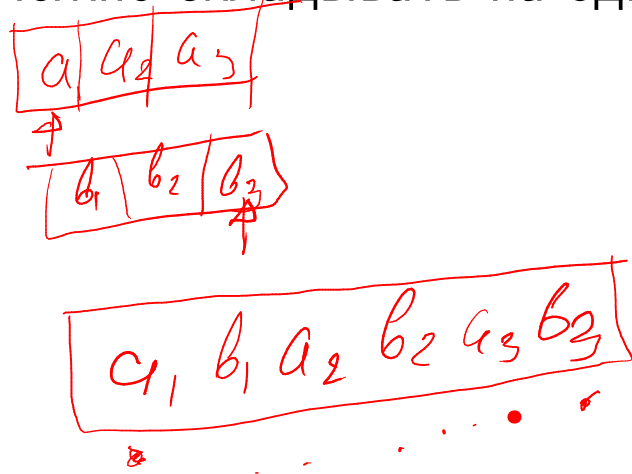
(Константа в $O(\dots)$ зависит только от размера записи машины M !)

Замечание

Можно промоделировать и на одноленточной машине из начала семестра, но замедление будет квадратичным.

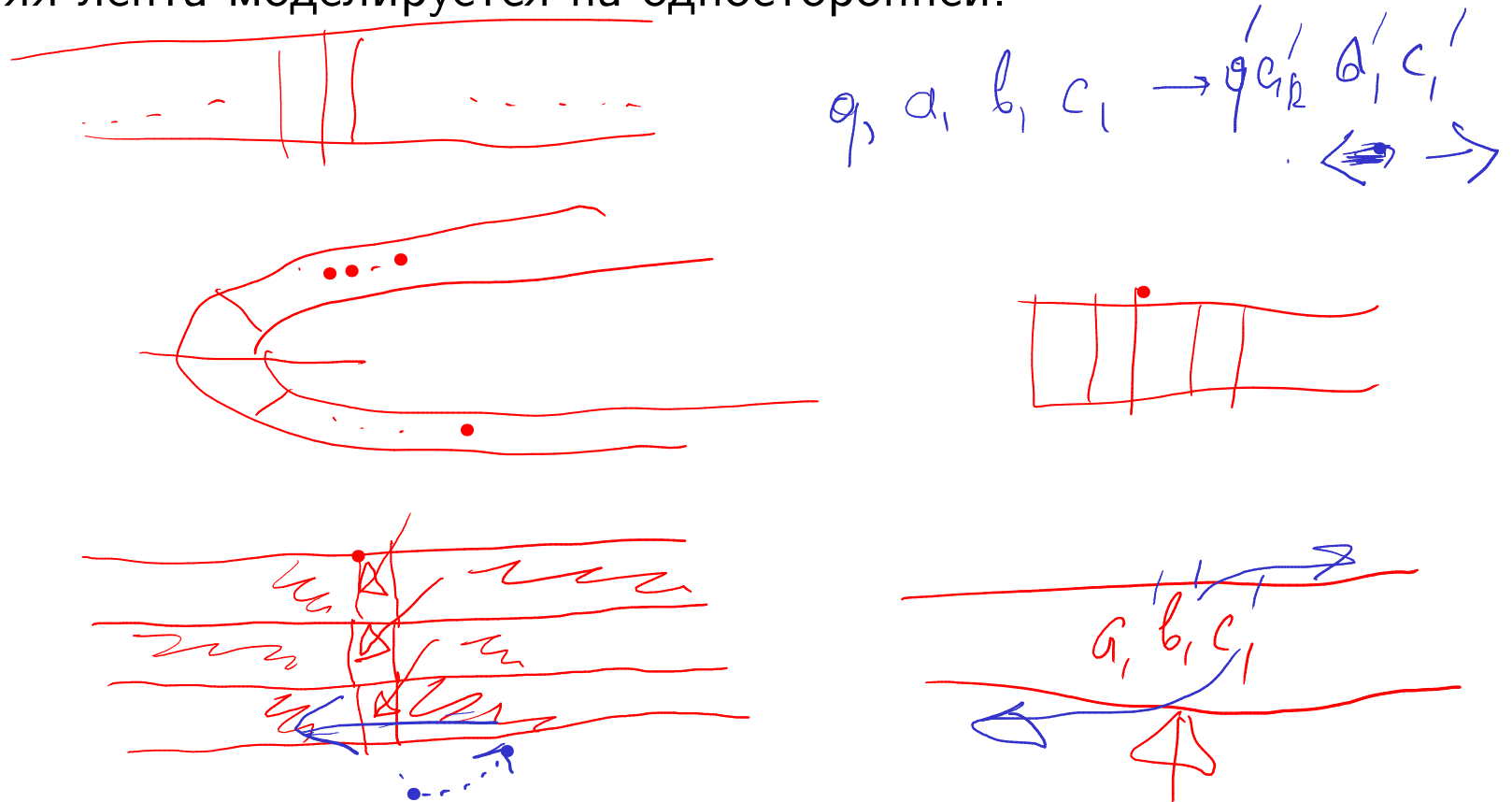
Двух лент достаточно: доказательство

- ▶ Много лент можно складывать на одной: через символ.



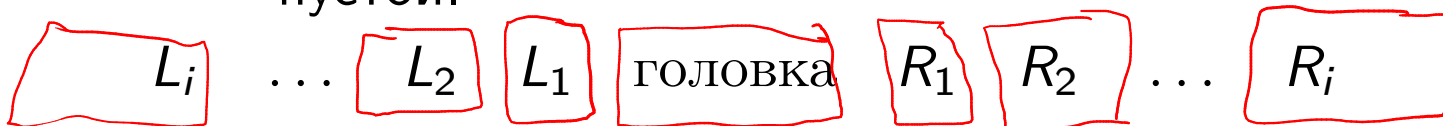
Двух лент достаточно: доказательство

- ▶ Много лент можно складывать на одной: через символ.
 - ▶ Головки можно совместить, и двигать ленты.
- Двусторонняя лента моделируется на односторонней.



Двух лент достаточно: доказательство

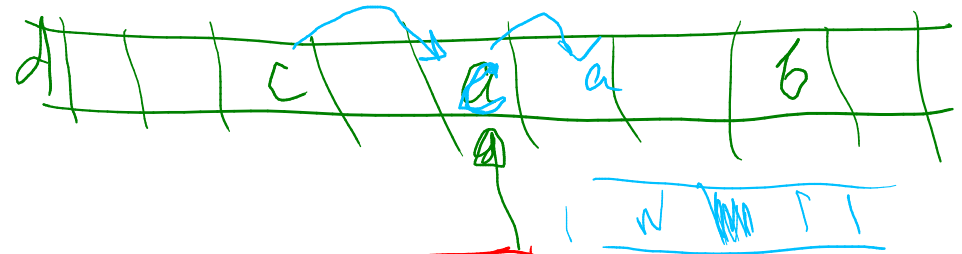
- ▶ Много лент можно складывать на одной: через символ.
- ▶ Головки можно совместить, и двигать ленты.
Двусторонняя лента моделируется на односторонней.
- ▶ Чтоб не двигать всю ленту, разделим на блоки по 2^i элементов, блок может быть:
 - ▶ заполнен полностью,
 - ▶ наполовину (резервные места!),
 - ▶ пустой.



Двух лент достаточно: доказательство

- ▶ Много лент можно складывать на одной: через символ.
- ▶ Головки можно совместить, и двигать ленты. Двусторонняя лента моделируется на односторонней.
- ▶ Чтоб не двигать всю ленту, разделим на блоки по 2^i элементов, блок может быть:

- ▶ заполнен полностью,
- ▶ наполовину (резервные места!),
- ▶ пустой.



- ▶ Инвариант: в L_i и R_i суммарно 2^i непустых пробелы не пустые! эл-тов.



Двух лент достаточно: доказательство

- ▶ Много лент можно складывать на одной: через символ.
- ▶ Головки можно совместить, и двигать ленты.
Двусторонняя лента моделируется на односторонней.
- ▶ Чтоб не двигать всю ленту, разделим на блоки по 2^i элементов, блок может быть:
 - ▶ заполнен полностью,
 - ▶ наполовину (резервные места!),
 - ▶ пустой.

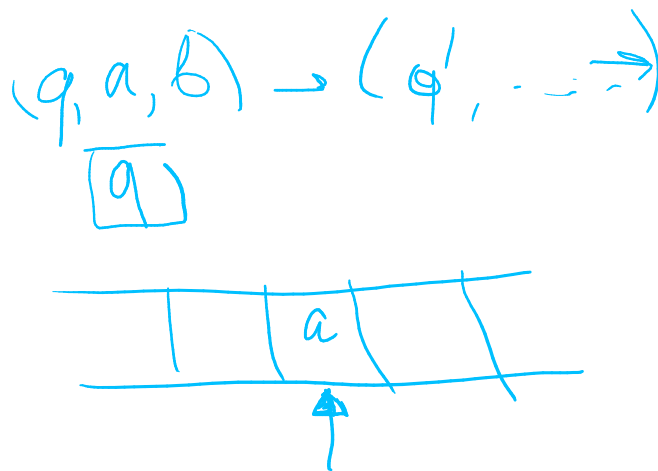


- ▶ Инвариант: в L_i и R_i суммарно 2^i непустых ^{пробелы не пустые!} эл-тов.
- ▶ Время работы:
 - ▶ если двигаем с блока i , то $C \cdot 2^i$ операций,
 - ▶ блок i трогаем не чаще, чем раз в 2^{i-1} шагов,
 - ▶ итого $\sum_i C 2^i \cdot \frac{t(\frac{n}{2^i})}{2^{i-1}} = O(t(\frac{n}{2}) \log t(\frac{n}{2}))$.

Универсальная машина Тьюринга

Теорема

\exists ДМТ U , выдающая на входе (M, x) тот же самый результат, что дала бы машина M на входе x , за время $O(t \log t)$, где t — время работы M на x .



Классы DTime и P

$t: \mathbb{N} \rightarrow \mathbb{N}$ называется **конструируемой по времени**, если

- ▶ $t(n) \geq n$,
- ▶ двоичную запись $t(|x|)$ можно найти по входу x на ДМТ за $t(|x|)$ шагов.

Язык $L \in \mathbf{DTime}[t(n)]$, если есть ДМТ M , принимающая L за время $O(t(n))$.

(Константа может зависеть от языка, но не от длины входа.)

$$\mathbf{P} = \bigcup_c \mathbf{DTime}[n^c]$$

Классы \widetilde{P} и \widetilde{NP}

Массовая задача R **полиномиально ограничена**, если существует полином p , ограничивающий длину кратчайшего решения:

$$\forall x (\exists u (x, u) \in R \Rightarrow \exists w ((x, w) \in R \wedge |w| \leq p(|x|))).$$

Массовая задача R **полиномиально проверяема**, если существует полином q , ограничивающий время проверки решения: для любой пары (x, w) можно проверить принадлежность $(x, w) \in R$ за время $q(|(x, w)|)$.

\widetilde{NP} — класс задач поиска, задаваемых полиномиально ограниченными полиномиально проверяемыми массовыми задачами.

Классы \widetilde{P} и \widetilde{NP}

Массовая задача R **полиномиально ограничена**, если существует полином p , ограничивающий длину кратчайшего решения:

$$\forall x (\exists u (x, u) \in R \Rightarrow \exists w ((x, w) \in R \wedge |w| \leq p(|x|))).$$

Массовая задача R **полиномиально проверяема**, если существует полином q , ограничивающий время проверки решения: для любой пары (x, w) можно проверить принадлежность $(x, w) \in R$ за время $q(|(x, w)|)$.

\widetilde{NP} — класс задач поиска, задаваемых полиномиально ограниченными полиномиально проверяемыми массовыми задачами.

\widetilde{P} — класс задач поиска из \widetilde{NP} , разрешимых за полиномиальное время, т.е. задаваемых отношениями R , такими, что $\forall x \in \{0, 1\}^*$ за полиномиальное время можно найти w , для которого $(x, w) \in R$.

Ключевой вопрос теории сложности: $\widetilde{P} \stackrel{?}{=} \widetilde{NP}$.

Классы P и NP

NP — класс языков (задач распознавания), задаваемых полиномиально ограниченными полиномиально проверяемыми массовыми задачами, т.е. $\mathbf{NP} = \{L(\underline{R}) \mid R \in \widetilde{\mathbf{NP}}\}$. $(x, w) \in R$

Иначе говоря, $L \in \mathbf{NP}$, если имеется п.о. п.п. R , такая, что

$$\forall x \in \{0, 1\}^* \quad x \in L \iff \exists w \ (\underline{x}, w) \in R.$$
 $L(R)$

P — класс языков (задач распознавания), распознаваемых за полиномиальное время; ясно, что $\mathbf{P} = \{L(R) \mid R \in \widetilde{\mathbf{P}}\}$.

Очевидно, $\mathbf{P} \subseteq \mathbf{NP}$.

Ключевой вопрос теории сложности: $\mathbf{P} \neq \mathbf{NP}$.