Структурная теория сложности

Эдуард Алексеевич Гирш

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

ПОМИ РАН

28 сентября 2008 г.

Напоминание: что мы решаем (1)

Работаем в алфавите $\{0,1\}$.
Множество слов длины n в нём: $\{0,1\}^n$.
Множество всех (конечных) слов: $\{0,1\}^*$.
Длина слова x: $|x|$.

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

$$\widetilde{\mathrm{FACTOR}} = \{(n, d) \mid n \vdots d,\ 1 < d < n\}.$$

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

$$\widetilde{\mathrm{FACTOR}} = \{(n, d) \mid n \vdots d,\ 1 < d < n\}.$$

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

Также решаем (распознаём) задачи распознавания (языки), т.е. задачи типа “да”/“нет”. Для массовой задачи R

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

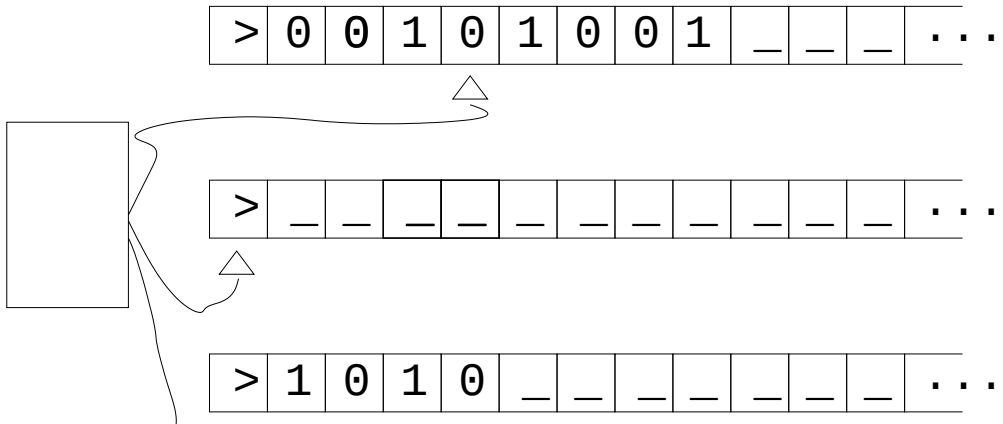
Например,

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

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 каждых $q, c_1, \ldots, c_k$ *одну* инструкцию вида $(q, c_1, \ldots) \mapsto (q', c'_1, \ldots, d_1, \ldots)$, где $q, q' \in Q$; $c_i, c'_i \in \Sigma$ — символы, обозреваемые головками; $d_i \in \{\leftarrow, \rightarrow, \cdot\}$ — направление движения.



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 первой 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. При сублинейных ограничениях на память первая лента (где вход) read-only и положение головки на ней не считается.

Классы $\widetilde{\mathbf{P}}$ и $\widetilde{\mathbf{NP}}$

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 $q(|(x, w)|)$.

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

Классы $\widetilde{\mathbf{P}}$ и $\widetilde{\mathbf{NP}}$

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 $q(|(x,w)|)$.

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

$\widetilde{\mathbf{P}}$ — класс задач поиска из $\widetilde{\mathbf{NP}}$, 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: $\widetilde{\mathbf{P}} \stackrel{?}{=} \widetilde{\mathbf{NP}}$.

Классы P и NP

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

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

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

P — класс задач распознавания (языков), разрешимых 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: $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$.

НМТ

Инструкции k-ленточной ДМТ можно записать как функцию (таблицу)

$$\delta\colon Q \times \Sigma^k \to Q \times \Sigma^k \times \{\leftarrow, \rightarrow, \cdot\}^k.$$

Недетерминированная машина Тьюринга (НМТ) допускает больше одной инструкции для данных $q \in Q$ и $c_1, \ldots, c_k \in \Sigma$, т.е. δ для неё — многозначная функция.

Так появляется дерево вычислений. . . [см. на доску]

В машины (ДМТ, НМТ) с заведомо ограниченным временем работы можно встроить будильник и считать время вычислений на входах одной длины всегда одним и тем же.

НМТ принимает вход, если $\exists$ путь в дереве вычислений, заканчивающийся q_Y.

НМТ и **NP** — другие определения

Недетерминированная машина Тьюринга (НМТ) — это просто ДМТ, у которой есть дополнительный аргумент (подсказка w на второй ленте).

НМТ M принимает вход x, если $\exists w$, для которой вычисление заканчивается в q_Y (пишем $M(x, w) = 1$).

Вычислительный путь в старом определении $\sim$ подсказка в новом.
Можно считать, что длина подсказки определяется длиной входа.

НМТ и **NP** — другие определения

Недетерминированная машина Тьюринга (НМТ) — это просто ДМТ, у которой есть дополнительный аргумент (подсказка w на второй ленте).

НМТ M принимает вход x, если $\exists w$, для которой вычисление заканчивается в q_Y (пишем $M(x, w) = 1$).

Вычислительный путь в старом определении $\sim$ подсказка в новом.
Можно считать, что длина подсказки определяется длиной входа.

Ещё одно определение **NP**:
NP — класс языков, принимаемых полиномиальными по времени НМТ.

Определения эквивалентны. . . [см. на доску].

Сведéния

Сведение языков по Карпу: $L_1 \longrightarrow L_2$, если
имеется полиномиально вычислимая f:

- $\forall x\ x \in L_1 \Leftrightarrow f(x) \in L_2$.

Сведение задач поиска по Левину: $R_1 \longrightarrow R_2$, если $\exists f, g, h\ \forall x_1, y_1, y_2$

- $R_1(x_1, y_1) \Rightarrow R_2(f(x_1), g(x_1, y_1))$,
- $R_1(x_1, h(f(x_1), y_2)) \Leftarrow R_2(f(x_1), y_2)$,
- f и h полиномиально вычислимы, а g ограничена полиномом.

Классы $\mathbf{P}$, $\mathbf{NP}$, $\widetilde{\mathbf{P}}$, $\widetilde{\mathbf{NP}}$ замкнуты относительно этих сведений.

Сведéния

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.

Сведение чего угодно по Тьюрингу: $A \longrightarrow B$, если имеется оракульная полиномиальная по времени машина $M^\bullet$, такая, что $A = L(M^B)$.

Сведéния

Сведение языков по Карпу: $L_1 \longrightarrow L_2$, если имеется полиномиально вычислимая f:

- $\forall x\ x \in L_1 \Leftrightarrow f(x) \in L_2$.

Сведение задач поиска по Левину: $R_1 \longrightarrow R_2$, если $\exists f, g, h\ \forall x_1, y_1, y_2$

- $R_1(x_1, y_1) \Rightarrow R_2(f(x_1), g(x_1, y_1))$,
- $R_1(x_1, h(f(x_1), y_2)) \Leftarrow R_2(f(x_1), y_2)$,
- f и h полиномиально вычислимы, а g ограничена полиномом.

Сведение чего угодно по Тьюрингу: $A \longrightarrow B$, если имеется оракульная полиномиальная по времени машина $M^{\bullet}$, такая, что $A = L(M^B)$.

Классы $\mathbf{P}$, $\widetilde{\mathbf{P}}$ замкнуты относительно этих сведений.
Классы $\mathbf{NP}$, $\widetilde{\mathbf{NP}}$ могут быть незамкнуты относительно сведений по Тьюрингу.

Трудные и полные задачи

Задача A — трудная для класса $\mathbf{C}$, если $\forall C \in \mathbf{C} \quad C \to A$.

Задача — полная для $\mathbf{C}$, если она трудная и принадлежит $\mathbf{C}$.

Теорема

Если

- A — **NP**-трудная,
- $A \in \mathbf{P}$,

то $\mathbf{P} = \mathbf{NP}$.

Следствие

Если A — **NP**-полная, то

$$A \in \mathbf{P} \Leftrightarrow \mathbf{P} = \mathbf{NP}.$$

NP-полная задача: ВН

Задача об ограниченной остановке:

$\widetilde{\text{ВН}}(\langle M, x, 1^t \rangle, w) = $ НМТ M с подсказкой w принимает вход x за $\leq t$ шагов

Теорема

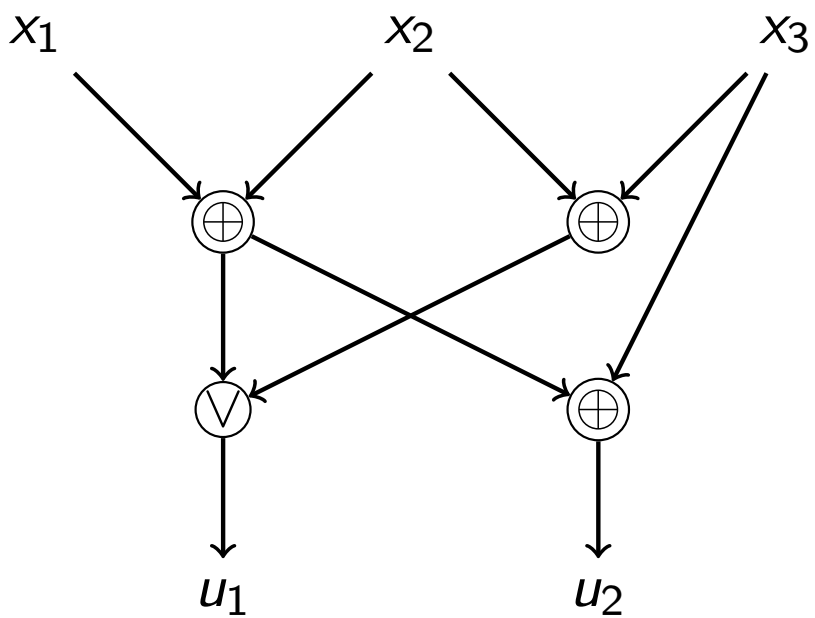
Задача об ограниченной остановке — $\widetilde{\mathbf{NP}}$-полная,
а соответствующий язык — **NP**-полный.

Замечание

Принадлежность $\widetilde{\mathbf{NP}}$ использует существование *универсальной* ДМТ, которая может промоделировать вычисление ДМТ, описание которой дано ей на вход. Причём с лишь линейным замедлением.

Булева схема

- Ориентированный граф без циклов.
- Бинарные (и унарные) операции над битами: $\wedge$, $\vee$, $\oplus$, ...
- Пример (4 гейта):



NP-полная задача: CIRCUIT_SAT

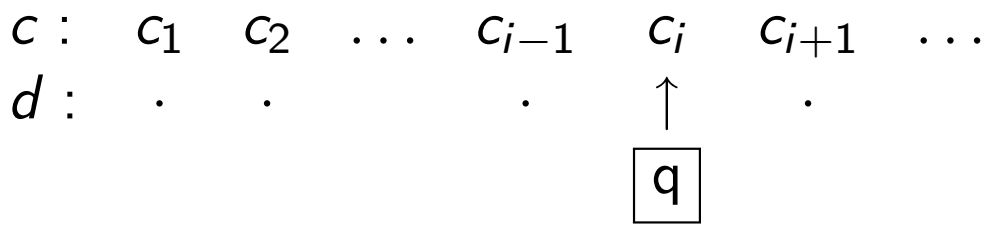
$\widetilde{\text{CIRCUIT\_SAT}} = \{(C, x) \mid C - \text{схема},\ C(x) = 1\}.$

ВН $\to$ CIRCUIT_SAT:

- этаж схемы — конфигурация ДМТ;
- время $t \Rightarrow t$ больших этажей схемы;
- время $t \Rightarrow t$ ячеек на этаже;
- переход между этажами реализует один шаг ДМТ:

$$(q', c_i', d_i') \leftarrow\!\shortmid (q, c_{i-1}, c_i, c_{i+1}, d_{i-1}, d_i, d_{i+1}),$$

где d_j, d_j' — «головка в j-й позиции»:



- входы схемы — подсказка НМТ (вход НМТ уже подставлен).
- выход схемы — попадание в q_Y.

NP-полная задача: 3-SAT

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

Пример:

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

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

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

- для последнего гейта g клоз (g).

$\widetilde{\mathbf{NP}}$ 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: $\widetilde{\mathrm{FACTOR}}$.

Не **NP**-полные задачи в **NP** $\setminus$ **P**

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

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

$f(n)$:

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

(II) за n шагов:

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

(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);

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

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

Оптимальный алгоритм для $\widetilde{\mathbf{NP}}$-задачи

- Перебираем все машины (не только полиномиальные).
- На шаге номер $i = 2^l(1 + 2k)$ моделируем k-ый шаг машины M_l.
- Если выдан ответ, проверяем его.
- Сравним со временем работы M_i:
  $t(x) \leq \mathrm{const}_i \cdot t_i(x) + p(|x|)$ (на проверку),
  если моделировать шаг-в-шаг;
- На ДМТ шаг будет стоить дорого, т.к. машин много
  одновременно, поэтому $t(x) \leq \mathrm{const}_i \cdot p(t_i(x)))$.