Лекция 12

Схемная сложность классов полиномиальной иерархии и PP

(Конспект: Р. Мясников, К. Первышев)

Доказать $\mathbf{NP} \not\subseteq \mathbf{P/poly}$ не проще, чем $\mathbf{NP} \neq \mathbf{P}$. Более того, до сих пор не удается доказать, что языки из $\mathbf{NP}$ не могут быть распознаны булевыми схем размера $O(n^2)$. Тем не менее, для некоторых более широких классов это можно доказать. Этим мы и займемся.

Определение 12.1. Обозначим $\mathbf{Size}[f(n)]$ класс языков, распознаваемых булевыми схемами размера $O(f(n))$.

12.1 Схемная сложность классов полиномиальной иерархии

Лемма 12.1. $\Sigma^4\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$ *для любого* k.

Доказательство. Количество схем размера n^k меньше количества функций, зависящих только от первых $c \cdot k \cdot \log n$ битов. Значит, существует подобная функция, не имеющая представления в виде булевой схемы размера n^k. Покажем, что одна из подобных функций лежит в $\Sigma^4\mathbf{P}$.

Заметим, что так как любая из рассматриваемых функций зависит от $c \cdot k \cdot \log n$ битов, то она может быть закодирована полиномиальным числом битов (например, в виде таблицы истинности). Невычислимость функции f с помощью булевой схемы размера n^k на входах длины n может быть записана следующим образом: $\forall$ схемы c (размера n^k) $\exists$ вход x (длины n), для которого $f(x) \neq c(x)$.

Введем на множестве рассматриваемых функций какой-нибудь просто вычислимый порядок (например, лексикографический). Теперь рассмотрим наименьшую по этому порядку невычислимую с помощью булевых схем размера n^k функцию f и определим язык L как множество слов, на которых эта функция принимает значение 1. Говоря строго,

$$y \in L \iff \exists f : \begin{cases} \forall c \exists x : f(x) \neq c(x) \\ \forall f' : (\forall c \exists x : f'(x) \neq c(x)) \implies f \leq f' \\ f(y) = 1 \end{cases}$$

Это выражение можно переписать как

$$\begin{aligned} y \in L \iff \exists f\ \forall c \forall f'\ \exists x \exists c'\ \forall x' : \\ f(x) \neq c(x) \wedge ((f \leq f') \vee f'(x') = c'(x')) \wedge f(y) = 1. \end{aligned}$$

Мы описали язык L из $\Sigma^4\mathbf{P}$, который не может быть описан схемой размера n^k. Но нас интересует такой язык, который не может быыть описан схемами размера $O(n^k)$.

Возьмём схему c размера $C \cdot n^{k-1}$. Для некоторого n_0 верно, что $C \cdot n_0^{k-1} < n_0^k$. Очевидно, существует вход x длины n_0, на котором $c(x)$ неверно решает задачу о принадлежности L. Следовательно, $L \notin \mathbf{Size}[n^{k-1}]$. $\square$

Замечание 12.1. Из этого не следует, что $\Sigma^4\mathbf{P} \not\subseteq \mathbf{P/poly}$.

Следствие 12.1. $\mathbf{PH} \not\subseteq \mathbf{Size}[n^k]$ *для любого* k.

Теперь мы в состоянии подобраться ближе к **NP**, доказав более сильное утверждение:

Теорема 12.1. $\Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P} \not\subseteq \mathbf{Size}[n^k]$ *для любого* k.

Доказательство. Предположим противное: пусть $\Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P} \subseteq \mathbf{Size}[n^k]$ для некоторого k. Тогда класс **NP** принимается схемами полиномиального размера. Это влечёт коллапс полиномиальной иерархии до $\Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P}$ (см. лекцию 6). Тем самым, $\mathbf{PH} = \Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P} \subseteq \mathbf{Size}[n^k]$. Противоречие. $\square$

Замечание 12.2. Аналогичным образом можно доказать более сильные утверждения. На самом деле, наличие схем полиномиального размера для класса **NP** влечёт коллапс полиномиальной иерархии не только до $\Sigma^2\mathbf{P} \cap \Pi^2\mathbf{P}$, но и до $\mathbf{ZPP}^{\mathbf{NP}}$, и даже ещё сильнее. Следовательно, $\mathbf{ZPP}^{\mathbf{NP}} \not\subseteq \mathbf{Size}[n^k]$ для любого k.

12.2 Схемная сложность PP

От классов полиномиальной иерархии перейдём к классу **PP**. Напомним, что в одной из предыдущих лекций было доказано следующее:

$$\boxed{\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{PP}} \text{ (теорема Тода)}}$$

В то же время, $\mathbf{P}^{\mathbf{PP}} = \mathbf{P}^{\sharp\mathbf{P}}$.

Из теоремы Шамира мы знаем, что **PSPACE** имеет интерактивный протокол с prover'ом из **PSPACE**. Покажем следующее:

Теорема 12.2. $\mathbf{P}^{\sharp\mathbf{P}}$ *имеет интерактивный протокол с prover'ом из* $\mathbf{P}^{\sharp\mathbf{P}}$.

Доказательство. Примем сторону verifier'а и опишем интерактивный протокол. Для проверки принадлежности слова x языку L, распознаваемому машиной, обладающей оракулом из $\sharp\mathbf{P}$ и работающей полиномиальное время $q(n)$, мы нуждается в содействии prover'а лишь для вычисления перманента[1] матрицы.

Все вычисления производятся над конечным кольцом $\mathbb{K}$, размер которого выберем позднее. Пару (A, b), составленную из матрицы A и скаляра b, будем называть *хорошей* , или *корректной* , если $\operatorname{perm} A = b$, и *плохой* в противном случае. Ответив скаляром b на очередной вопрос о перманенте некоторой матрицы A, prover будет пытаться нас убедить в том, что пара (A, b) хорошая.

Разложим перманент матрицы $A = (a_{i,j})_{1 \le i,j \le n}$ по первой строке и получим $\operatorname{perm} A = \sum_{j=1}^{n} a_{1,j} \cdot \operatorname{perm} A_j$. Попросим prover для каждого j прислать $b_j = \operatorname{perm} A_j$. Проверим равенство $b = \sum_{j=1}^{n} a_{1,j} \cdot b_j$. Это обеспечит наличие хотя бы одной плохой пары среди пар (A_j, b_j) в случае, если пара (A, b) была плохой.

Сформируем набор $U = \{(A_j, b_j)\}_{j=1}^{n}$. Пока в наборе U содержится хотя бы два элемента, будем повторять следующую операцию. Возьмём из U две любые пары (D, d) и (E, e). Сформируем матрицу $C(x) = x \cdot D + (1 - x) \cdot E$. Заметим, что её перманент является многочленом $p(x)$, степень которого не больше n. Попросим prover прислать коэффициенты этого многочлена. Используя интерполяцию, хороший prover из класса $\mathbf{P}^{\sharp\mathbf{P}}$ это сделать в состоянии.

Итак, prover прислал коэффициенты некоторого многочлена $\tilde{p}$. Проверим, что $\tilde{p}(0) = e$ и $\tilde{p}(1) = d$. Это влечёт отличие многочлена $\tilde{p}$ от

[1] Перманент матрицы $A = (a_{i,j})_{1 \le i,j \le n}$ определяется как $\operatorname{perm} A = \sum_{i \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$, где S_n — группа всех перестановок $1..n$.

многочлена $p = \operatorname{perm} C(x)$ в случае, если хотя бы одна из пар (D, d) и (E, e) была плохой. Выберем случайным образом элемент r нашего кольца $\mathbb{K}$, и заменим эти две пары в наборе U на одну пару $(C(r), \tilde{p}(r))$. При условии $p \neq \tilde{p}$, r является корнем ненулевого полинома $p - \tilde{p}$, степень которого не более n, с вероятностью не более $\frac{n}{|\mathbb{K}|}$. Тем самым, если набор U содержал плохую пару, то и после описанного изменения с вероятностью как минимум $1 - \frac{n}{|\mathbb{K}|}$ в нём присутствует хотя бы одна плохая пара.

После повторения этой операции не более n раз, набор U будет состоять из одной-единственной пары (A', b'). Если исходная пара (A, b) была плохой, то и (A', b') плоха с вероятностью не менее $1 - \frac{n^2}{|\mathbb{K}|}$. Причём размер матрицы A' на единицу меньше размера исходной матрицы A. Таким образом задачу проверки корректности пары с матрицей размера n мы свели к аналогичной задаче с матрицей размера $n - 1$. И, тем самым, мы умеем проверять перманент матрицы A размера n с вероятностью ошибки не более $\frac{n^3}{|\mathbb{K}|}$.

В ходе работы prover'а количество запросов и размеры матриц, для которых вычисляется перманент, не превосходят $q(n)$. И вероятность, с которой verifier поверит обманщику, сказав "да" на $x \notin L$, не превышает $\frac{q(n)^3}{|\mathbb{K}|}$. Отсюда легко определить подходящий размер кольца $\mathbb{K}$. В то же время, хороший prover всегда сможет убедить verifier'а. □

Определим протоколы типа Мерлин-Артур. Как водится, Мерлин всемогущ, а Артур справедлив:

Определение 12.2 (MA, MA$_2$ — протоколы Мерлин-Артур). Язык $L \in \mathbf{MA}$, если существуют такие полиномы p и q, а также машина M, работающая на всех входах полиномиальное время, что для всех x верно следующее:

$$x \in L \implies \exists y \in \{0,1\}^{p(|x|)} : \Pr_{z \in \{0,1\}^{q(|x|)}} \{M(x, y, z) = 1\} > 3/4,$$
$$x \notin L \implies \forall y \in \{0,1\}^{p(|x|)} : \Pr_{z \in \{0,1\}^{q(|x|)}} \{M(x, y, z) = 1\} < 1/4.$$

Говоря неформально, Мерлин присылает доказательство, а Артур, подбросив несколько раз монетку, проверяет его.

Можно показать (см. следующую лекцию), что можно требовать, чтобы при $x \in L$ вероятность ошибки была равна нулю. Этот (несимметричный) вариант будем обозначать $\mathbf{MA}$, а симметричный — $\mathbf{MA}_2$.

Замечание 12.3.

Аналогично можно определить и протоколы типа Артур-Мерлин, в начале которого Артур сообщает Мерлину исходы подбрасывания монеток.

Определение 12.3 (AM — протоколы Артур-Мерлин). Язык $L \in \mathbf{AM}$, если существуют полиномы p и q, а также машина M, работающая на всех входах полиномиальное время, что для всех x верно следующее:

$$x \in L \implies \Pr_{z\in\{0,1\}^{q(|x|)}}\{\exists y \in \{0,1\}^{p(|x|)} : M(x,y,z) = 1\} > 3/4$$
$$x \notin L \implies \Pr_{z\in\{0,1\}^{q(|x|)}}\{\exists y \in \{0,1\}^{p(|x|)} : M(x,y,z) = 1\} < 1/4$$

Упражнение 12.1. $\mathbf{MA}_2 \subseteq \mathbf{AM}$.

Упражнение 12.2. В предположении, что $\mathbf{NP} \subseteq \mathbf{P/poly}$, имеет место $\mathbf{MA} = \mathbf{AM}$.

Лемма 12.2. *В предположении, что* $\mathbf{PP} \subseteq \mathbf{P/poly}$*, имеет место* $\mathbf{P^{PP}} \subseteq \mathbf{MA}$.

Доказательство. Поскольку каждая машина из $\mathbf{P^{PP}}$ задаёт вопросы к оракулу из $\mathbf{PP}$ не более чем полиномиальной длины, имеем $\mathbf{P^{\sharp P}} = \mathbf{P^{PP}} \subseteq \mathbf{P/poly}$, где подсказками являются наборы схем, вычисляющие ответы оракула из класса $\mathbf{PP}$ на входах длины не более $\mathrm{poly}(n)$.

Возьмём интерактивный протокол для класса $\mathbf{P^{\sharp P}}$ с prover’ом из $\mathbf{P^{\sharp P}}$. Потребуем, чтобы prover не помнил истории диалога с verifier’ом, но verifier при каждом своём обращении к prover’у отсылал бы эту историю (имеющую полиномиальную длину). Ясно, что при наложении таких требований хороший prover будет по-прежнему в состоянии доказать принадлежность слова языку, а плохой prover будет всё так же пойман с большой вероятностью. Тем самым, хороший prover превращается в обычную машину из $\mathbf{P^{\sharp P}}$.

Теперь позовём Артура с Мерлином. Артур будет моделировать verifier, но вместо обращений к prover’у Артур будет использовать схемы, присланные Мерлином в самом начале выполнения протокола. Эти схемы будут полноценным заменителем prover’а, поскольку запросы verifier’а имеют длину $\mathrm{poly}(n)$, а $\mathbf{P^{\sharp P}} \subseteq \mathbf{P/poly}$. Таким образом, $\mathbf{P^{PP}} = \mathbf{P^{\sharp P}} \subseteq \mathbf{MA}$. □

Перейдём непосредственно к интересующему нас утверждению:

Теорема 12.3. $\mathbf{PP} \not\subseteq \mathbf{Size}[n^k]$ *для любого* k.

Доказательство. В предположении, что для некоторого k выполняется $\mathbf{PP} \subseteq \mathbf{Size}[n^k]$, имеем:

$$\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{PP}} \subseteq \mathbf{MA}_2 \subseteq \mathbf{PP},$$

(последнее включение $\mathbf{MA}_2 \subseteq \mathbf{PP}$ — техническая лемма 12.3), откуда в силу $\mathbf{PH} \not\subseteq \mathbf{Size}[n^k]$ имеем $\mathbf{PP} \not\subseteq \mathbf{Size}[n^k]$. □

Лемма 12.3. $\mathbf{MA}_2 \subseteq \mathbf{PP}$.

Доказательство. Пусть $L \in \mathbf{MA}_2$. Чтобы уменьшить вероятность ошибки Артура, выполним его процедуру над одним и тем же доказательством Мерлина несколько раз, каждый раз подкидывая монетки заново. Выберем ответ, встречающийся наибольшее число раз, и воспользуемся неравенством Чернова подобно тому, как это делалось для **BPP**.

Таким образом, имеются такие полиномы p и q, а также машина M, работающая на всех входах полиномиальное время, что для всех x верно следующее:

$$x \in L \implies \exists y \in \{0,1\}^{p(|x|)} : \Pr_{z \in \{0,1\}^{q(|x|)}} \{M(x,y,z) = 1\} > 1 - 4^{-p(|x|)}$$
$$x \notin L \implies \forall y \in \{0,1\}^{p(|x|)} : \Pr_{z \in \{0,1\}^{q(|x|)}} \{M(x,y,z) = 1\} < 4^{-p(|x|)}$$

Будем выбирать y случайным образом (вместо того, чтобы спрашивать его у Мерлина) — тем самым, мы рассматриваем последовательности подбрасываний монеток $(y,z) \in \{0,1\}^{p(|x|)+q(|x|)}$. Тогда

$$x \in L \implies \Pr_{(y,z)} \{M(x,y,z) = 1\} > 2^{-p(|x|)} \cdot (1 - 4^{-p(|x|)}) > 4^{-p(|x|)}$$
$$x \notin L \implies \Pr_{(y,z)} \{M(x,y,z) = 1\} < 4^{-p(|x|)}$$

Заметим, что порог $1/2$ в определении класса **PP** может быть успешно заменён на любую функцию вида $a(x) \cdot 2^{-s(|x|)}$, где s — полином, а a — функция, вычислимая за полиномиальное время. В частности, на $4^{-p(|x|)}$. □