Лекция 9

Теорема Тода́ (вторая часть)

(Конспект: Д. Ицыксон)

Итак, нам осталось доказать, что $\mathbf{BPP}^{\oplus\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{PP}}$. Очевидно, что $\mathbf{BPP}^{\oplus\mathbf{P}} \subseteq \mathbf{PP}^{\oplus\mathbf{P}}$. Значит, нам достаточно показать включение $\mathbf{PP}^{\oplus\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{PP}}$.

Пусть M — НМТ. Обозначим через $acc_M(x)$ количество принимающих путей машины M на входе x (иначе говоря, количество подсказок, приводящих M на x в принимающее состояние).

Определение 9.1. $\#\mathbf{P} = \{f : \Sigma^* \to \mathbb{N} \cup \{0\} \mid \exists$ полиномиальная по времени НМТ M_f, такая, что $\forall x\ f(x) = acc_{M_f}(x)\}$.

Таким образом, $\#\mathbf{P}$ — это обобщение $\mathbf{NP}$. В частности, в $\#\mathbf{P}$ имеется задача $\#\mathtt{SAT}$, в которой по булевой формуле требуется определить *количество* выполняющих ее наборов.

Утверждение 9.1. $\mathbf{P}^{\#\mathbf{P}} = \mathbf{P}^{\mathbf{PP}}$.

Доказательство. 1. $\mathbf{P}^{\mathbf{PP}} \subseteq \mathbf{P}^{\#\mathbf{P}}$. Не умаляя общности, можно считать, что все ветви дерева вычислений НМТ - одинаковой длины. С помощью оракула мы можем узнать, сколько принимающих вычислений у той или иной машины Тьюринга, нас интересует больше ли это половины всех. Посчитаем это на самой машине.

2. $\mathbf{P}^{\#\mathbf{P}} \subseteq \mathbf{P}^{\mathbf{PP}}$. Пусть наш оракул из $\#\mathbf{P}$ задан НМТ M. Определим язык $L = \{(x \in \Sigma^*, y \in \mathbb{N}) : acc_M(x) > y\}$. Если мы покажем, что $L \in \mathbf{PP}$, тогда двоичным поиском за полиномиальное время можно будет найти точное значение $acc_M(x)$: всего вычислений $2^{\mathrm{poly}(n)}$, а двоичный поиск работает за $\log$, то есть за $\mathrm{poly}(n)$.

Построим НМТ M', показывающую, что $L \in \mathbf{PP}$. Первый ее шаг: недетерминированный выбор, в одной ветви запускается машина M, в другой происходят фиктивные вычисления, которых столько же, сколько у машины M, но из них в отвергающее состояние попадает ровно y веток. Этого легко добиться, смотря только на двоичную запись подсказки (меньше она числа y или больше).

Сколько принимающих путей у M'? Их $acc_M(x) + (all_M - y)$, где all_M – это полное количество путей машины M. Для нахождения вероятности принятия это число нужно поделить на $2 \cdot all$. Получим $P\{\text{принятия } (x,y)\} = \frac{1}{2} + \frac{acc_M(x)-y}{2 \cdot all}$, что больше $\frac{1}{2}$ тогда и только тогда, когда $acc_M(x) > y$. □

Лемма 9.1. *Семейство функций s_i задано следующим рекуррентным соотношением: $s_i(z) = 3(s_{i-1}(z))^4 + 4(s_{i-1}(z))^3$, $s_0(z) = z$. Тогда $\forall i \geq 0\ \forall z \in \mathbb{N}$ верно следующее: если $2|z$, то $2^{2^i}|s_i(z)$, иначе $2^{2^i}|(s_i(z)+1)$.*

Упражнение 9.1. Доказать лемму 9.1 по индукции. □

Теорема 9.1. $\mathbf{PP}^{\oplus\mathbf{P}} \subseteq \mathbf{P}^{\#\mathbf{P}}$.

Доказательство. Пусть $L \in \mathbf{PP}^{\oplus\mathbf{P}}$. Тогда

$$L = \{x \,|\, \exists A \in \mathbf{P}^{\oplus\mathbf{P}} = \oplus\mathbf{P} : |\{y \in \{0,1\}^{p(|x|)} | (x,y) \in A\}| > \frac{1}{2}2^{p(|x|)}\},$$

где p — полином. В дальнейшем $l(x) := \lceil \log p(|x|) + 1 \rceil$.

Покажем, что $L \in \mathbf{P}^{\#\mathbf{P}}$, для этого опишем машину M (это будет большой «монстр») следующим образом.

1. Вычисляем коэффициенты $q(z) = (s_{l(x)}(z))^2$ рекурсивно. Очевидно, что мы вычислим их за полиномиальное время; значит, количество ненулевых коэффициентов будет также полиномиально. На каждом шаге рекурсии степень возрастает не больше, чем в 4 раза. Т.е. степень этого полинома не более $16^{l(x)}$, что меньше, чем $p^4(|x|)$.

2. Производим недетерминированный выбор, так чтобы веток было столько же, сколько ненулевых коэффициентов многочлена $q(z)$. В каждой из этих веток производится расщепление еще на несколько веток, именно, на количество, равное соответствующему коэффициенту $q(z)$, а затем в каждой такой веточке последовательно запускается НМТ $\mathcal{A}$, соответствующая языку A, столько раз, какова степень при соответствующем коэффициенте. Причем запускаем следующую машину, только если предыдущая закончила работу в принимающем состоянии.

У машины M ровно $(s_{l(x)}(z))^2$ принимающих вычислений, где z — это количество принимающих вычислений машины A на входе (x,y). Если $(x,y) \in A$, то у машины $\mathcal{A}$ было нечетное число принимающих вычислений, тогда по лемме 9.1 у M их будет 1 по модулю $2^{2^{l(x)}}$. Если не принадлежит, то 0 по тому же модулю.

Мы не можем спрашивать оракула о количестве принимающих путей машины M для каждой подсказки y (поскольку подсказок экспоненциально много), поэтому мы объединим деревья вычислений этих машин для разных y в одно. Итак, мы опишем машину N, количество принимающих путей которой мы будем спрашивать у оракула, чтобы выяснить, верно ли, что $x \in L$. Нам хотелось оценить количество $y \in \{0,1\}^{p(|x|)}$, для которых $(x,y) \in A$; вернее, знать, больше ли их, чем $\frac{1}{2}2^{p(|x|)}$. Наша машина N недетерминированно угадывает подсказку y, затем запускает $M(x,y)$. Так как число подсказок y равно $2^{p(|x|)} < 2^{2^{l(x)}} \leq 2^{p(|x|)+1}$, то число нужных подсказок y — это как раз количество принимающих путей машины N по модулю $2^{2^{l(x)}}$. Осталось это число сравнить с $\frac{1}{2}2^{p(|x|)}$. $\square$

Итак, теорема Тодá доказана: мы доказали

$$\mathbf{PH} \subseteq \mathbf{BPP}^{\oplus\mathbf{P}} \subseteq \mathbf{PP}^{\oplus\mathbf{P}} \subseteq \mathbf{P}^{\#\mathbf{P}} = \mathbf{P}^{\mathbf{PP}}.$$

Упражнение 9.2. Доказать, что $\mathbf{PP}^{\mathbf{PH}} \subseteq \mathbf{P}^{\mathbf{PP}}$. Указание: для этого достаточно обобщить лемму про $\mathbf{BPP}^{\mathbf{BPP}^A} \subseteq \mathbf{BPP}^A$ до $\mathbf{PP}^{\mathbf{BPP}^A} \subseteq \mathbf{PP}^A$. $\square$