Лекция 13

$\mathbf{BPP} \subseteq \exists \cdot \mathbf{BPP} \subseteq \mathbf{NP}^{\mathbf{BPP}} \subseteq \mathbf{MA}_2 = \mathbf{MA} \subseteq \mathbf{ZPP}^{\mathbf{NP}} \subseteq \Sigma_2 \cap \Pi_2$

(Конспект: М. Гладких, К. Ушаков)

Определение 13.1. $L \in \exists \cdot \mathcal{C}$, если существует язык $L' \in \mathcal{C}$ такой, что

$$L = \{\, x \mid \exists y\ (x, y) \in L' \,\}.$$

Доказательство

Для начала отметим несколько тривиальных включений:
1)$\mathbf{BPP} \subseteq \exists \cdot \mathbf{BPP}$
2)$\exists \cdot \mathbf{BPP} \subseteq \mathbf{NP}^{\mathbf{BPP}}$
3)$\mathbf{ZPP}^{\mathbf{NP}} \subseteq \Sigma_2 \cap \Pi_2$

Замечание 13.1. 1-ое и 2-ое включения прямо следуют из определения оператора Шонинга $\exists \cdot \mathbf{BPP}$. Для доказательства 3-го включения достаточно заметить, что $\mathbf{RP}^{\mathbf{NP}} \subseteq \mathbf{NP}^{\mathbf{NP}}$, а по определению $\mathbf{ZPP} = \mathbf{RP} \cap \mathbf{co\text{-}RP}$.

Утверждение 13.1. $\mathbf{NP}^{\mathbf{BPP}} \subseteq \mathbf{MA}_2$.

Доказательство. Машина из $\mathbf{NP}$ имеет дерево вычислений полиномиальной высоты $p(n)$. В этом дереве мы осуществляем вызовы оракула из $\mathbf{BPP}$ (будем считать, что вероятность ошибки в $\mathbf{BPP}$ меньше $\frac{1}{2^n}$).

Пусть Мерлин дает Артуру путь в этом дереве.

Если $x \in L$, тогда такой путь существовал, а значит Мерлин может показать правильный путь. Артур может его попытаться проверить, вычисляя ответ оракула самостоятельно (случайные числа для этого у него имеются). При проверке правильного пути Артур может его отвергнуть из-за ошибочного ответа при вычислении ответа оракула из **BPP**. Так как вероятность ошибки алгоритма в **BPP** экспоненциально мала, то и вероятность того, что мы пропустим неправильное доказательство, тоже мала (к оракулу мы могли успеть обратиться лишь полиномиальное число раз).

Если $x \notin L$, то Мерлин дал неверный путь и Артур, проверяя его, либо дойдет до конца и отвергнет его, либо получит ошибку при вычислении ответа оракула из **BPP**, вероятность чего опять мала. □

Утверждение 13.2. $\mathbf{MA}_2 = \mathbf{MA}$.

Доказательство. Вложение $\mathbf{MA} \subseteq \mathbf{MA}_2$ тривиально, докажем обратное включение.

Все множество строк случайных строк делится на хорошие и плохие (Артур на них врет). Плохих строчек мало. Тогда попытаемся сдвинуть множество хороших строк так, чтобы покрыть все множество.

Формально, хотим, чтобы для языка $L \in \mathbf{MA}_2$ было выполнено ($\oplus$ это XOR):

1) если $x \in L$, то $\exists w\ \exists r_1, \dots, r_k\ \forall r \bigvee_{i=1}^{k} A(x, w, r \oplus r_i) = 1$, то есть хотя бы одна из строк $r \oplus r_i$ — хорошая (w — доказательство Мерлина из $\mathbf{MA}_2$);

2) если $x \notin L$ то $\forall w\ \forall r_1, \dots, r_k\ \Pr\left\{\bigvee_{i=1}^{k} A(x, w, r \oplus r_i) = 1\right\} \leq \frac{1}{3}$, то есть количество случайных строк, опровергающих доказательство Мерлина, останется большим.

Ясно, что из этого следует $\mathbf{MA}_2 \subseteq \mathbf{MA}$: выше написано в точности определение **MA**. Остается доказать, что 1) и 2) выполнено для любого языка $L \in \mathbf{MA}_2$.

Будем считать, что вероятность ошибки Артура — $\frac{1}{2^u}$, причем u выберем позднее.

Сначала докажем существование $r_1, \dots, r_k$. Временно фиксируем r. Выберем строки $r_1, \dots, r_k$ случайным образом и посчитаем вероятность $\Pr\{A(x, w, r \oplus r_i) = 0\}$. Это означает, что все r_i плохие для исходного Артура; вспомним, что одна r_i была плохой с вероятностью $\frac{1}{2^u}$, а их k штук. Тогда при фиксированном r получим $\Pr\left\{\bigvee_{i=1}^{k} A(x, w, r \oplus r_i) = 0\right\} \leq \left(\frac{1}{2^u}\right)^k$. Итого $\Pr\left\{\exists r\ \bigvee_{i=1}^{k} A(x, w, r \oplus r_i) = 0\right\} \leq \frac{2^{|r|}}{2^{ku}}$. А тогда нам достаточно выбрать u, k такими, что $\frac{2^{|r|}}{2^{ku}} < 1$, после этого автоматически получим существование $r_1, \dots, r_k$, ведь мы их выбрали случайно, и получили, что вероятность успешного набора больше 0. Выберем u, k чуть позже.

Пусть $x \notin L$. Мы взяли строку r, и при сдвиге мы получили плохую строку, то есть r лежало в множестве, являющемся сдвигом плохого множества подсказок на r_i, его доля от всех подсказок $\frac{1}{2^u}$. Тогда мы хотим, чтобы $\frac{k}{2^u} < \frac{1}{3}$.

Возьмем $u = |x| + |w|$ и $k = |r|$, легко заметить, что при этом выполняются оба требуемых неравенства.

□

Утверждение 13.3. $\mathbf{MA} \subseteq \mathbf{ZPP^{NP}}$

Доказательство. Построим машину из $\mathbf{ZPP^{NP}}$ (т.е. вероятностную оракульную машину Тьюринга и оракул из $\mathbf{NP}$), которая принимает заданный язык из $\mathbf{MA}$.

Шаг 1. Пусть вероятность ошибки Артура — $\frac{1}{2^u}$ (u мы выберем позже). Выберем $r_1, \ldots, r_{2u}$ случайным образом. Спрашиваем у оракула из $\mathbf{NP}$: существует ли w, такое, что $\bigwedge_{i=1}^{2u} A(x, w, r_i) = 1$. Если оракул сказал “нет”, тогда пусть наша машина говорит “нет”, если же оракул сказал “да”, то продолжаем строить алгоритм.

Ответ “нет”, выданный на этом шаге, всегда верный. В самом деле, если $x \in L$, то Мерлин знал такое w, а значит, оракул не мог сказать “нет”.

Пусть $x \notin L$. Посчитаем, с какой вероятностью оракул мог сказать “да”: $\Pr\{\exists w \ldots\} \leq 2^{|w|} \Pr\{\bigwedge_{i=1}^{2u} A(x, w, r_i) = 1 \text{ для конкретного } w\} \leq$
$\leq 2^{|w|} \left(\Pr\{A(x, w, r) = 1\}\right)^{2u} \leq 2^{|w|} \left(\frac{1}{2^u}\right)^{2u} \ll \frac{1}{2}$ (этого можно добиться выбором u).

Шаг 2. Теперь явно установим w. Мы сделаем это побитово. Так как оракул сказал “да”, то спросим у оракула: существует ли $\widetilde{w}$, такое, что $\bigwedge_{i=1}^{2u} A(x, \widetilde{w}, r_i) = 1$, и ее первый бит “0”. Если оракул сказал “да”, то спросим про первые два бита “00”, если же оракул сказал “нет”, то спросим про “10”, и так далее… Таким образом считаем, что после первого шага, если такое w существует, то мы его знаем.

Далее, если мы успешно прошли первый шаг, случайно выберем новые строки $s_1, \ldots, s_k$, и спросим у оракула: для любого ли r выполнено $\bigvee_{i=1}^{k} A(x, w, r \oplus s_i) = 1$. Если оракул сказал “да”, тогда пусть наша машина говорит “да”, если же оракул сказал “нет”, то наша машина говорит “не знаю”.

Ответ “да”, выданный на этом шаге, всегда верный. В самом деле, если $x \notin L$, то обязательно существует такая случайная строчка $\widetilde{r}$, для которой $\bigvee_{i=1}^{k} A(x, w, \widetilde{r} \oplus s_i) = 0$, так как иначе мы бы покрыли все множество случайных строк полиномиальным числом экземпляров множества ошибочных строк (имеющего лишь экспоненциально малую долю).

В каких случаях мы говорим "не знаю"? Если на первом шаге мы пропустили $x \notin L$ (вероятность $\ll \frac{1}{2}$, как мы выяснили), или же мы сказали "не знаю" на $x \in L$. Какова вероятность второго события?

Покажем, что если доказательство w подходит для $< 1/4$ доли случайных строк, то вероятность того, что нам его дадут на первом шаге, мала. Действительно, вероятность того, что конкретное w подошло на первом шаге, не превосходит $\left(\frac{1}{4}\right)^{2u}$; тогда вероятность того, что оракул выдал нам какое-то плохое w, не превосходит $2^{|w|}\left(\frac{1}{4}\right)^{2u}$.

Если же w подходит для $\geq 1/4$ доли случайных строк, то аналогично утверждению 13.2 можно доказать, что для некоторого k вероятность того, что $\bigvee_{i=1}^{k} A(x, w, r \oplus s_i) = 0$, меньше $\frac{1}{4}$: действительно $\Pr\{\bigvee_{i=1}^{k} A(x, w, r \oplus s_i) = 0\} \leq (\Pr\{r \oplus s_1 \notin A\})^k \leq (3/4)^k \ll 1/4$.

Таким образом, вероятность того, что мы скажем "не знаю" на $x \in L$, много меньше $\frac{1}{2}$.

Значения для u и k, достаточные для того, чтобы используемые нами оценки выполнялись, найти несложно. $\square$