Рябых Владислав, Б05-905

Алгоритмы и модели вычислений. Домашнее задание № 9

Задача 1. Докажите, что $\mathcal{RP} \subseteq \mathcal{NP}$.

Решение. Будем работать с представлением ВМТ $M(x)$, как МТ с дополнительной лентой случайных битов, которая читается слева направо. Рассмотрим произвольный язык $L \in \mathcal{RP}$, произвольное слово $x \in L$.
По определению класса $\mathcal{RP}$: $x \in L \Rightarrow \mathbb{P}\{M(x) = 1\} \geqslant \frac{1}{2}$, тогда существует, обрабатываемая МТ часть ленты случайных битов q, подав которую в качестве сертификата для слова x недетерминированной машине Тьюринга $A(x, q)$ (которая строится как исходная ВМТ $M(x)$, только вместо ленты случайных битов имеет ленту, с которой читает сертификат), мы получим единицу: $A(x, q) = 1$. Заметим, что так как в определении класса $\mathcal{RP}$ МТ полиномиальная, то, приняв слово x, использованная при этом часть ленты также имеет полиномиальную от размера входа длину: $q = poly(|x|)$. Также, так как по определению класса $\mathcal{RP}$: $x \notin L \Rightarrow \mathbb{P}\{M(x) = 0\} = 1$, то на любом сертификате s верно, что $\forall s\ \forall x \notin L \hookrightarrow A(x, s) = 0$.
Таким образом получаем, что $x \in L \Leftrightarrow \exists q = poly(|x|) : A(x, q) = 1$, где $A(x, q)$ вычислима за $poly(|x|)$. Что и есть определение класса $\mathcal{NP}$, так что $L \in \mathcal{NP}$, а следовательно $\mathcal{RP} \subseteq \mathcal{NP}$.

Задача 2. 1. Докажите, что если $\mathcal{P} = \mathcal{NP}$, то $\mathcal{P} = \mathcal{BPP}$.

2. Докажите, что если $\mathcal{NP} \subseteq$ co-$\mathcal{RP}$, то $\mathcal{ZPP} = \mathcal{NP}$.

Решение. 1.
- Из семинара мы знаем, что $\mathcal{P} \subseteq \mathcal{RP}$; а по предыдущей задаче, что $\mathcal{RP} \subseteq \mathcal{NP}$. Тогда, в силу предположения $\mathcal{P} = \mathcal{NP}$, получаем $\mathcal{P} = \mathcal{RP} = \mathcal{NP}$. Тогда $\mathcal{P} = \mathcal{RP} \subseteq \mathcal{BPP}$.
- Также мы знаем, что $\mathcal{BPP} \subseteq \Pi_2 \cap \Sigma_2$. Тогда $\mathcal{BPP} \subseteq \mathcal{PH}$. Однако, как мы знаем, $\mathcal{P} = \mathcal{NP} \Leftrightarrow \mathcal{P} = \mathcal{NP} = \mathcal{PH}$, то есть отсюда получаем $\mathcal{BPP} \subseteq \mathcal{P}$.

Таким образом, из предположения $\mathcal{P} = \mathcal{NP}$ следует $\mathcal{P} = \mathcal{BPP}$.

2. $\mathcal{ZPP} \subseteq \mathcal{RP}$ по своему определению, а $\mathcal{RP} \subseteq \mathcal{NP}$ по первой задаче. Тогда, если $\mathcal{NP} \subseteq$ co-$\mathcal{RP}$, то $\mathcal{ZPP} \subseteq \mathcal{RP} \subseteq \mathcal{NP} \subseteq$ co-$\mathcal{RP}$.

   В таком случае $L \in \mathcal{RP} \Rightarrow L \in$ co-$\mathcal{RP} \Rightarrow \overline{L} \in \mathcal{RP} \Rightarrow \overline{\overline{L}} = L \in \mathcal{RP}$. Тогда co-$\mathcal{RP} \subseteq \mathcal{RP}$ и получаем, что $\mathcal{RP} = \mathcal{NP} =$ co-$\mathcal{RP}$, то есть $\mathcal{ZPP} \subseteq \mathcal{RP} \cap$ co-$\mathcal{RP} = \mathcal{NP}$

   Осталось доказать, что $\mathcal{NP} = \mathcal{RP} \cap$ co-$\mathcal{RP} \subseteq \mathcal{ZPP}$. Возьмём произвольный язык $L \in \mathcal{RP}$. Для того, чтобы доказать, что $L \in \mathcal{ZPP}$ будем запускать ВМТ как для L, так и для $\overline{L}$. В таком случае ВМТ A проверяет, что $x \in L$ (и безошибочно определяет, что $x \notin L$), а ВМТ B, что $x \in \overline{L}$ (и, таким образом, безошибочно определяет, что $x \in L$). Тогда мы будем запускать эти 2 ВМТ, пока одна из них не выдаст 0 в то время как другая выдаст 1 (в таком случае мы безошибочно можем сказать, что если $A(x) = 1 \wedge B(x) = 0$, то $x \in L$ и, если $A(x) = 0 \wedge B(x) = 1$, то $x \notin L$). Матожидание количества запусков равняется 2 (так как вероятность остановиться есть $\frac{1}{2}$), так что матожидание времени работы полиномиально.

Таким образом получаем, что $\mathcal{NP} = \mathcal{RP} \cap$ co-$\mathcal{RP} \subseteq \mathcal{ZPP}$, чего нам и не хватало для доказательства $\mathcal{ZPP} = \mathcal{NP}$.

Задача 3. Покажите, что в задаче сравнения больших файлов, разобранной на семинаре, вероятность ошибки действительно не превосходит $3/4$ при достаточно больших n. Оцените, насколько должно быть велико n и покажите, что n бит $\geqslant 32$ мегабайта — достаточное количество для справедливости оценок.

Решение. Оценка с семинара на количество «плохих» простых чисел: $k \leqslant \ln 2 \cdot \dfrac{n}{\ln n}$.

Оценка с семинара на количество простых чисел: $0.99 \cdot \dfrac{k}{\ln k} \leqslant prime(k) \leqslant 1.01 \cdot \dfrac{k}{\ln k}$.

Тогда на $[n, 2n]$ как минимум $0.99 \cdot \dfrac{2n}{\ln 2n} - 1.01 \cdot \dfrac{n}{\ln n}$ простых чисел и, таким образом, вероятность выбрать «плохое» простое число $P \leqslant \dfrac{\ln 2 \cdot \dfrac{n}{\ln n}}{0.99 \cdot \dfrac{2n}{\ln 2n} - 1.01 \cdot \dfrac{n}{\ln n}}$

Тогда $P \leqslant \dfrac{3}{4} \Rightarrow \dfrac{\dfrac{\ln 2}{\ln n}}{\dfrac{2 \cdot 0.99}{\ln 2n} - \dfrac{1.01}{\ln n}} \leqslant \dfrac{3}{4} \Rightarrow n \geqslant 5.1 \cdot 10^{12}$

Однако 32 мегабайта это приблизительно $2.6 \cdot 10^8$ бит, так что наша оценка оказалась слишком грубой. Возьмём тогда в качестве оценки на количество простых чисел $0.999 \cdot \dfrac{k}{\ln k} \leqslant prime(k) \leqslant 1.0045 \cdot \dfrac{k}{\ln k}$.

Теперь получим $P \leqslant \dfrac{3}{4} \Rightarrow \dfrac{\dfrac{\ln 2}{\ln n}}{\dfrac{2 \cdot 0.999}{\ln 2n} - \dfrac{1.0045}{\ln n}} \leqslant \dfrac{3}{4} \Rightarrow n \geqslant 2.4 \cdot 10^8$

Так что $n \geqslant 32$ мегабайта $\approx 2.6 \cdot 10^8$ бит действительно достаточно

Задача 4. Покажите, что класс $\mathcal{BPP}$ не изменится, если

1. константу стандарта Монте-Карло $\frac{1}{3}$ заменить на любое число, строго меньшее $\frac{1}{2}$
2. полиномиальное в среднем число шагов заменить на полиномиальное число шагов.

Решение. 1. пусть новая константа C ($C \in [0, \frac{1}{2})$), а класс, получившийся при замене исходной константы стандарта Монте-Карло $\frac{1}{3}$ на C, обозначим как $\mathcal{BPP}_C$.

- Рассмотрим, случай с $C \leqslant \frac{1}{3}$: очевидно, $\mathcal{BPP}_C \subseteq \mathcal{BPP}$. Обратное включение получим, нужное количество раз (пока вероятность ошибки не уменьшится до значения C, само количество можно оценить с помощью) запустив исходную ВМТ для языка $\mathcal{BPP}$ и выбрав наиболее часто встречающийся ответ (что по факту есть функция большинства).
- Случай с $C \in [\frac{1}{3}, \frac{1}{2})$ аналогично: $\mathcal{BPP} \subseteq \mathcal{BPP}_C$. Обратное включение получим, нужное количество раз (пока вероятность ошибки не уменьшится до значения C, само количество можно оценить с помощью) запустив исходную ВМТ для языка $\mathcal{BPP}_C$ и выбрав наиболее часто встречающийся ответ (что по факту есть функция большинства).

2. Обозначим получившийся класс как $\mathcal{BPP}'$. Очевидно, что полиномиальное число шагов также полиномиально в среднем, то есть $\mathcal{BPP}' \subseteq \mathcal{BPP}$. Докажем обратное включение: пусть алгоритм работает в среднем $t(|x|)$ тактов. Мы же будем останавливать его через, допустим, $8 \cdot t(|x|)$ тактов. Тогда по неравенству Маркова вероятность того, что мы прервались и выдали неправильный ответ: $\mathbb{P}\{\text{неправильный} \mid \text{прервались}\} = \dfrac{t(|x|)}{8 \cdot t(|x|)} = \dfrac{1}{8}$.

   $\mathbb{P}\{\text{неправильный} \mid \text{не прерывались}\} = \dfrac{1}{3}$.

   Таким образом:

   $\mathbb{P}\{\text{неправильный}\} \leqslant \mathbb{P}\{\text{неправильный} \mid \text{прервались}\} + \mathbb{P}\{\text{неправильный} \mid \text{не прерывались}\} = \dfrac{1}{3} + \dfrac{1}{8} = \dfrac{11}{24} < \dfrac{1}{2}$.

   По предыдущему пункту мы можем заменить исходную константу на любую константу C, такую что $C \in [0, \frac{1}{2})$, так что получаем: $\mathcal{BPP} \subseteq \mathcal{BPP}_C = \mathcal{BPP}'$, чего нам и не хватало для доказательства $\mathcal{BPP}' = \mathcal{BPP}$

Задача 5. Задача 2 из приложенного файла (пункты (i) и (iv)).

Решение. (i): $(AB - C)x = 0 \Leftrightarrow \forall k \in \overline{1, n} \hookrightarrow \sum_{i=1}^{n} \alpha_{ki} \cdot x_i = 0$

Компоненты вектора x выбираются из множества $\{0, 1, \ldots, N-1\}$, наш полином имеет степень 1. Тогда по лемме Шварца-Зиппеля вероятность того, что на данном наборе компонент полином обращается в 0 не превышает $\dfrac{1}{N}$. Тогда вероятность того, что все полиномы системы обращаются в 0 (а это и есть ошибка в нашей задаче), не превышает $\dfrac{1}{N^n}$. Тогда, если мы хотим, чтобы вероятность ошибки была меньше заданной вероятности p, то $\dfrac{1}{N^n} < p \Rightarrow N > p^{-\frac{1}{n}}$

(iv): $(A(Bx)x) = (Cx)x \Leftrightarrow ((AB)x)x - (Cx)x = 0 \Leftrightarrow ((AB - C)x)x = 0$

Это многочлен 2 степени от переменных x_k, так что по лемме Шварца-Зиппеля вероятность обращения в 0 не превышает $\dfrac{2}{N}$. Тогда $\dfrac{2}{N} < p \Rightarrow N > \dfrac{2}{p}$

$(A(Bx)y) = (Cx)y \Leftrightarrow ((AB)x)y - (Cx)y = 0 \Leftrightarrow ((AB - C)x)y = 0$

Теперь это многочлен 2 степени от $2n$ переменных, которые, впрочем, выбираются также из множества $\{0, 1, \ldots, N-1\}$, так что по лемме Шварца-Зиппеля вероятность обращения в 0 не превышает $\dfrac{2}{N}$. Тогда $\dfrac{2}{N} < p \Rightarrow N > \dfrac{2}{p}$

Задача 7. Докажите, что 2-CNF — задача из $\mathcal{P}$. В какой из вероятностных классов, определенных выше, попадает язык 2-CNF?

Решение. Во 2м номере домашнего задания на шестую неделю было доказано, что 2-$CNF \in \mathcal{NL}$. Но, как мы знаем, $\mathcal{NL} \subseteq \mathcal{P}$, так что 2-$CNF \in \mathcal{P}$.

Если 2-КНФ выполнима, то МТ останавливается за полиномиальное от длины входа время. Пусть матожидание времени работы есть $t(|x|)$. Аналогично задаче 4, будем останавливать алгоритм и выдавать, что $x \notin$ 2-CNF через $8 \cdot t(|x|)$ тактов. Тогда, если $x \in$ 2-CNF, то вероятность ошибки не превышает $\dfrac{1}{2}$, а если $x \notin$ 2-CNF, то мы не ошибаемся вовсе.

Таким образом, 2-$CNF \in \mathcal{RP} \subseteq \mathcal{BPP}$.