

# Зачетные задачи

Зотов Алексей 497

May 27, 2017

**Задача. 1.** Построить систему интерактивных доказательств для языка **GI-NO-EQUAL-CLASSES**  $= \{(G_1, \dots, G_m) \mid \text{в разбиении этого набора графов на классы эквивалентности по отношению изоморфизма нет двух классов одинакового размера}\}$

**Ответ.** Мы уже знаем, что  $\mathbf{GNI} \in \mathbf{IP}$  и будем это использовать. Также  $\mathbf{GI} \in \mathbf{NP}$ .  $M = \{1, \dots, m\}$   
Рассмотрим такой протокол :

1.  $\forall i \in 1, \dots, m$  верификатор  $V$  посылает пруверу  $P$  индекс  $i$  соответствующий  $G_i$ .
2.  $P$  возвращает  $X_i = \{(k, S_{ki}) \mid G_k \cong G_i\}$  - множество индексов графов, изоморфных  $G_i$  и соответствующие сертификаты изоморфности.  $X_i = (K_i, S_i)$  - обозначение.
3.  $V$  проверяет полученные сертификаты.
4.  $\forall j : j \notin K_i$  верификатор  $V$  инициирует протокол проверки  $(G_i, G_j) \in \mathbf{GNI}$ , с вероятностью ошибки  $p_{ij} \leq \frac{1}{3}$ .
5. Алгоритм повторяется с пункта (1), игнорируя те индексы, для которых уже найден класс изоморфности.
6.  $V$  проверяет, что все классы получились разного размера. Возвращает **True**, если на каждый шаг протокола корректный (проверка сертификатов, проверка на  $G_i \not\cong G_j$ ), и полученные классы разного размера. Иначе **False**.

Докажем, что алгоритм корректен:

- если  $X = (G_1, \dots, G_m) \in \mathbf{GI-NO-EQUAL-CLASSES}$ , тогда каждый на каждой итерации прувер будет действовать корректно, положительная проверка на изоморфность и неизоморфность проходит без ошибок (с вероятностью 1).
- если  $X = (G_1, \dots, G_m) \notin \mathbf{GI-NO-EQUAL-CLASSES}$ , тогда  $P$  не может неизоморфные графы отнести в один класс (т.к. проверка сертификатов детерминированная), но может попробовать изоморфные графы разбить по разным классам, воспользовавшись наличием ошибки при проверке  $G_i \not\cong G_j$ . На каждой такой проверке вероятность обмануть верификатор  $p_{ij} \leq \frac{1}{3}$ . Значит вероятность ошибочно принять  $X$ :  $P_{\text{err}} \leq \frac{1}{3}$ .

**Задача. 4.** Постройте систему интерактивных доказательств с общими случайными битами для языка  $\mathbf{GROUP-NI} = \{G_0, G_1 \mid G_0, G_1 - \text{таблицы умножения двух неизоморфных конечных групп}\}$

**Ответ.** Проверить, что данные таблицы это таблицы умножения групп, верификатор может без прувера за  $O(n^2)$ . Достаточно проверить ассоциативность, наличие единицы и обратимость всех элементов. Нужно проверить их неизоморфность.

Рассмотрим  $S = \{(H, \sigma) \mid H \cong G_i, i \in \{0, 1\}, \sigma \in \text{Aut} H\}$ . Тогда, если  $G_0 \cong G_1$ , то  $|S| = n!$ , иначе  $|S| = 2 \cdot n!$ . Каждую группу порядка  $n$  можно записать двоичным числом длины  $m$ , где  $m = p(n)$ . Обозначим  $K = 2n!$ , имеем  $S \subset \{0, 1\}^m$ . Выберем  $k$  таким, что  $2^{k-2} \leq K \leq 2^{k-1}$ . Рассмотрим такой протокол:

- $V$  выбирает случайную хэш-функцию  $h$  из семейства попарно независимых полиномиально вычислимых (от  $m, k$ ) хеш-функций  $H_{m,k} : 2^m \rightarrow 2^k$ . Также выбирает случайный  $y \in 0, 1^k$ . Отправляет пруверу  $P$  пару  $(h, y)$  (значит можно считать, что случайные биты - общие).

- $P$  выбирает  $x \in S : h(x) = y$ . Возвращает верификатору пару  $(x, s)$ , где  $s$  - сертификат  $x \in S$ .
- $V$  проверяет сертификат  $s$  и  $h(x) = y$ . Принимает доказательство если проверки корректные.

Покажем корректность протокола. Пусть  $p = \frac{|S|}{2^k}$ .

- $P_{h,y}\{\exists x \in S : h(x) = y\} \leq p$  - так как  $|h(S)| \leq |S|$ .
- Рассмотрим  $E_x$  - событие  $\{h(x) = y\}$ .  $Pr\{\cup_{x \in S} E_x\} \geq \sum_{x \in S} Pr\{E_x\} - \sum_{x < x' \in S} Pr\{E_x \cap E_{x'}\} = \frac{|S|}{2^k} - \frac{|S|(|S|-1)}{2} \frac{1}{2^{2k}} > p(1 - \frac{p}{2}) \geq \frac{3}{4}p$  (так как  $p \leq \frac{1}{2}$  из-за выбора  $p$  и ограничения сверху на размер  $S$ ).

Получили  $\frac{3}{4}p \leq P_{h,y}\{\exists x \in S : h(x) = y\} \leq p$ , значит :

- Если  $|S| \geq K$ , то  $\frac{3}{4}p_0 \leq P_{h,y}\{\exists x \in S : h(x) = y\}$
- Если  $|S| \leq \frac{K}{2}$ , то  $P_{h,y}\{\exists x \in S : h(x) = y\} \leq p \leq \frac{p_0}{2} < \frac{3}{4}p_0$

Для разных случаев получили некоторый вероятностный зазор, который можно увеличить полиномиальным числом повторений протокола.

**Задача. 2.** Пусть есть  $m = n(n-1)/2$  булевых схем полиномиального размера  $\phi_1, \dots, \phi_m$  со входом длины  $k$  и одним выходом. Для каждого  $x \in \{0, 1\}^k$  рассмотрим граф  $G_x$  на  $n$  вершинах, матрица смежности которого задана результатами работы схем  $\phi_j$  на входе  $x$ . Рассмотрим множество графов  $\mathcal{G} = \{G_x | x \in \{0, 1\}^k\}$ . Пусть мы хотим отделить наборы  $\phi_1, \dots, \phi_m$ , когда в множестве  $\mathcal{G}$  менее  $C$  различных попарно неизоморфных графов от наборов, когда их хотя бы  $D$  ( $D \geq C$ ). Существует ли интерактивная система доказательств, которая делает это при  $C = D$ ? Можете ли вы её построить? Если не можете, то попробуйте её построить для случая  $C = D/2$ .

**Ответ.** Да, такой протокол существует.  $\mathbf{L} = \{(\phi_1 \dots \phi_m)\}$  : в  $\mathcal{G}$  содержится менее  $D$  попарно неизоморфных графов}.  $\mathbf{L} \in \mathbf{PSAPCE}$ . Можно посчитать количество классов изоморфности графов на полиномиальной памяти, для этого храним текущее число классов, для каждого не рассмотренного графа  $G_i$ , перебираем все уже рассмотренные  $G_j$  (перебор графов, или наборов булевых формул это одно и то же), если  $\forall j \leq i : G_j \not\cong G_i$ , то увеличиваем число классов эквивалентности. Полученное число классов эквивалентности и будет ответом. Дальше сравним его с  $D$ . Значит  $\mathbf{L} \in \mathbf{PSAPCE}$ . Значит  $\mathbf{IP} = \mathbf{PSAPCE}$ . Значит нужный протокол из  $\mathbf{IP}$  существует.

Для  $C = D/2$  подойдет протокол подробно описанный выше, в задаче (4).

**Задача. 3.** Пусть  $S \in \mathbf{NP}$ . Обозначим через  $S_n$  множество  $S \cap \{0, 1\}^n$ . Постройте систему интерактивных доказательств, получающую на вход число  $K$ , такую что если  $|S_n| > K$ , то пружер убеждает верификатора с вероятностью 1 (а не  $2/3$ , как на лекции), а если  $|S_n| < K/2$ , то пружер убеждает с вероятностью не больше  $1/3$ . Можно ли заменить  $K/2$  на  $0.99K$ ? (Аргумент полинома во времени работы верификатора - это  $n$ ).

**Ответ.** Используем протокол, подробно описанный в задаче (4).

Для  $0.99K$  достаточно проверить не размер множества  $S$ , а размер множества  $(S \times S \times \dots \times S) = S^l$ , которое очевидно лежит в  $\mathbf{NP}$ ,  $l$  выбираем так, что  $0.99^l \leq \frac{1}{2}$ . Получаем сравнение для размеров  $M$  и  $K$ , где  $\frac{1}{2}K^l \leq M \leq K^l$ .

Ошибку первого рода можно до 0, так как мы получили протокол из  $\mathbf{IP}$ , для которого в одном из эквивалентных определений соответствующая ошибка равна 0.

**Задача. 5.** Определим класс  $\mathbf{AMA}'$  и  $\mathbf{AMA}''$  так:  $B \in \mathbf{AMA}'(\mathbf{AMA}'')$ , если существует полиномиальный алгоритм  $V(x, r, s, q)$ , такой что:

- Если  $x \in B$ , то  $\Pr_r[\exists s \Pr_q[V(x, r, s, q) = 1] \geq \frac{2}{3}] \geq \frac{2}{3}$
- (для  $\mathbf{AMA}'$ ) Если  $x \notin B$ , то  $\Pr_r[\exists s \Pr_q[V(x, r, s, q) = 1] \geq \frac{2}{3}] \leq \frac{1}{3}$
- (для  $\mathbf{AMA}''$ ) Если  $x \notin B$ , то  $\Pr_r[\forall s \Pr_q[V(x, r, s, q) = 1] \leq \frac{1}{3}] \geq \frac{2}{3}$

(а) (1 балл) Поясните, в чём отличие трёх определений. А именно, почему один и тот же  $V$  может удовлетворить одному и не удовлетворить другому.

(б) (4 балла) Докажите, что  $\mathbf{PP} \subset \mathbf{AMA}'$ .

(в) (5 баллов) Докажите, что  $\mathbf{AMA}'' = \mathbf{AMA}$ .

**Ответ.** 1.  $\forall s \mathbf{Pr}_q[V(x, r, s, q) = 1] \leq \frac{1}{3} \Leftrightarrow \nexists s \mathbf{Pr}_q[V(x, r, s, q) = 1] > \frac{1}{3}$ .  
получаем для **АМА'** и **АМА''** соответственно :

- (для **АМА'**) Если  $x \notin B$ , то  $\mathbf{Pr}_r[\exists s \mathbf{Pr}_q[V(x, r, s, q) = 1] \geq \frac{2}{3}] \leq \frac{1}{3}$
- (для **АМА''**) Если  $x \notin B$ , то  $\mathbf{Pr}_r[\nexists s \mathbf{Pr}_q[V(x, r, s, q) = 1] > \frac{1}{3}] \geq \frac{2}{3}$

Пусть тогда  $V$  такой, что  $x \in B \implies \forall r \exists s \mathbf{Pr}_q[V(x, r, s, q) = 1] = \frac{1}{2}$ , тогда :

$\mathbf{Pr}_r[\exists s \mathbf{Pr}_q[V(x, r, s, q) = 1] \geq \frac{2}{3}] = 0 \leq \frac{1}{3}$  - выполнено

$\mathbf{Pr}_r[\nexists s \mathbf{Pr}_q[V(x, r, s, q) = 1] > \frac{1}{3}] = 0 < \frac{2}{3}$  - не выполнено

2. Пусть  $L \in \mathbf{PP}$ , значит  $\exists M$  :

- $x \in L \implies P_q[M(x, q) = 1] \geq \frac{1}{2}$
- $x \notin L \implies P_q[M(x, q) = 1] < \frac{1}{2}$

Заметим, что  $\frac{1}{2}$  из определения можно заменить на произвольную константу  $\in (0, 1)$ . Это можно сделать, например, добавив некоторое количество случайных бит, и на некотором фиксированном числе случайных исходов (фиксированных для каждого  $n$ ) выдавать ответ True, вне зависимости от входа. Это увеличит константу в определении. Воспользуемся определением с константой  $\frac{2}{3}$ .

Положим тогда  $V(x, s, r, q) = M(x, q) \quad \forall r, s$ . Тогда :

- $x \in L \implies P_r[\exists s : P_q[V(x, r, s, q) = M(x, q) = 1] \geq \frac{2}{3}] = 1 > \frac{2}{3}$
- $x \notin L \implies P_r[\exists s : P_q[V(x, r, s, q) = M(x, q) = 1] \geq \frac{2}{3}] = 0 < \frac{2}{3}$

Значит  $L \in \mathbf{АМА}'$ .

3. (a) Пусть  $L \in \mathbf{АМА}$ . Есть протокол: общие случайные биты  $r$  и  $q \rightarrow$  Артур получает  $r \rightarrow$  Мерлин возвращает  $s$ , зная  $r \rightarrow$  Артур получает случайные биты  $q$  to Выдает вердикт  $V(x, r, s, q)$ . Так как значение констант в определении **АМА** не играет, будем считать их достаточно близкими к 0 и к 1 соответственно ( $\varepsilon, 1 - \varepsilon$ ). Тогда :

- Выполнено :  $x \in L \implies \mathbf{Pr}_r[\exists s \mathbf{Pr}_q[V(x, r, s, q) = 1] \geq \frac{2}{3}] \geq \frac{2}{3}$ , иначе  $P_{r,q}(V(x, r, s, q) = 1) < \frac{2}{3} + \frac{1}{3}\varepsilon = \frac{8}{9} < 1 - \varepsilon$ . (Здесь считаем что  $P$  выбирает наилучшее  $s$ ).
- Выполнено :  $x \notin L \implies \mathbf{Pr}_r[\forall s \mathbf{Pr}_q[V(x, r, s, q) = 1] \leq \frac{1}{3}] \geq \frac{2}{3}$ , иначе  $P_{r,q}(V(x, r, s, q) = 1) > \frac{2}{3} + \frac{1}{3}\varepsilon = \frac{7}{9} > \varepsilon$

Значит  $L \in \mathbf{АМА}''$ .

(b) Пусть теперь  $L \in \mathbf{АМА}''$ , значит существует соответствующая  $V$ . Протокол будет как в пункте 1, но изменим процедуру принятия доказательства. Пусть Артур получает сразу несколько групп случайных битов  $q_1, q_2, \dots, q_m$  перед принятием решения. Рассмотрим такую процедуру:

- Артур запускает  $V(x, r, s, q_1)$  и  $V(x, r, s, q_2)$ , если получил в обоих случаях TRUE, то принимает, если в обоих случаях FALSE, то отвергает, иначе повторяет сначала, используя новые случайные биты. Так повторяет некоторое количество раз (подробнее ниже), если так и не получил ответ, то выбирает случайно FALSE или TRUE.

Посмотрим на вероятности:

- Пусть  $x \in L$ , тогда Артур с вероятностью (по  $r$ )  $\geq \frac{2}{3}$  попадет в благоприятный случай, где  $\exists s$ , который и выберет Мерлин, такой, что  $\mathbf{Pr}_q[V(x, r, s, q) = 1] \geq \frac{2}{3}$  (тут  $q = q_1$  или  $q_2$ ). Тогда, применив процедуру описанную выше, Артур с вероятностью  $p_1 \geq \frac{4}{9}$  примет доказательство на 1 шаге, с вероятностью  $p'_1 \leq \frac{1}{9}$  отвергнет, в остальных случаях он повторит.

Если бы он повторял бесконечное число раз, то вероятность принять  $p = p_1 + cp_1 + c^2p_1 + \dots = p_1(1 + c + c^2 + \dots) = p_1C$ , отверг с  $p' = p'_1 + cp'_1 + c^2p'_1 + \dots = p'_1(1 + c + c^2 + \dots) = p'_1C$ . Но  $p_1C + p'_1C = p + p' = 1 \implies C = \frac{1}{p_1 + p'_1}$ . Получим, что  $p \geq \frac{4}{5}$ . Понятно, что если мы применим меньшее число раз то получим немного худшую оценку, но последовательность  $C$  - геометрическая прогрессия, поэтому можно взять полином повторений и приблизиться экспоненциально. (можно даже константу раз, т.к. нам нужно константное приближение например с точностью возможной ошибкой  $\delta$ ). Значит итоговая вероятность принять  $x$  не меньше  $\frac{2}{3}(\frac{4}{5} - \delta) = \frac{8}{15} - \frac{2\delta}{3} > \frac{1}{2} + \delta_0$ .

- Пусть  $x \notin L$ , пользуемся вторым пунктом определения  $АМА''$ . Тогда Артур с вероятностью  $\leq \frac{1}{3}$  попадет в случай, где машина ошибаться может часто. С вероятностью  $\geq \frac{2}{3}$  попадает где ошибок малая доля (из условия). Там процедура принятия аналогичная, значит вероятность принять слово  $x : p \leq \frac{1}{5} + \delta$  получается аналогично предыдущему пункту. Итого вероятность ошибочно принять  $x : P \leq \frac{1}{3} + \frac{2}{3}(\frac{1}{5} + \delta) = \frac{7}{15} + \frac{2\delta}{3} < \frac{1}{2} - \delta_0$ . Получили разделимую (на  $2\delta_0$ ) границу.  $L \in АМА''$ .

**Задача. 6.** Пусть  $G$  является генератором псевдослучайных чисел. Рассмотрим следующие модификации:

- $G'(s) = \begin{cases} 0^{|G(s)|}, & \text{если } s \text{ содержит ровно } \frac{|s|}{2} \text{ единиц} \\ G(s), & \text{иначе} \end{cases}$
- $G'(s) = \begin{cases} 0^{|G(s)|}, & \text{если } s \text{ содержит ровно } \frac{|s|}{3} \text{ единиц} \\ G(s), & \text{иначе} \end{cases}$

Какие из этих функций являются генераторами псевдослучайных чисел и почему?

**Ответ.** Считаем  $n = |s|$ . В обоих случаях полиномиальная вычислимость  $G'(s)$  очевидна. Нужно проверить пункт (2) определения.

1.

$$G'(s) = \begin{cases} 0^{|G(s)|}, & \text{если } s \text{ содержит ровно } \frac{|s|}{2} \text{ единиц} \\ G(s), & \text{иначе} \end{cases} \quad (1)$$

$G'(s)$  - не является ГПСЧ.

В  $s$  ровно  $\frac{|s|}{2}$  единиц в  $C_n^{\frac{n}{2}}$  различных  $s$ . Считая, что  $s \sim U_n$  и воспользовавшись тем, что для достаточно больших  $n$  выполнено  $C_n^{\frac{n}{2}} > \frac{2^n}{n+1}$ , получим:

$$P(G(s) = 0^{p(n)}) \geq \frac{C_n^{\frac{n}{2}}}{2^n} \geq \frac{1}{n+1} \quad n \geq N_0 \quad (2)$$

Воспользуемся определением вычислительной неотличимости,  $y_n \sim U_{p(n)}$ , пусть  $\{D_n\}$  - такое семейство схем, что  $D_n(x) = 1 \iff x = 0^n$ . Получим :  
 $|P\{D_n(G'(s)) = 1\} - P\{D_n(y_n) = 1\}| \geq \frac{1}{n+1} - \frac{1}{2^n} \geq \frac{1}{2(n+1)}$ . при  $n \geq 10$ . Также  $\frac{1}{2(n+1)} \geq \frac{1}{2(p(n)+1)}$  при  $n > N_p$ . То есть мы получили, что  $\exists \{D_n\}, \exists q(p(n)) = \frac{1}{2(p(n)+1)} \forall N \exists n > N : |P\{D_n(G'(s)) = 1\} - P\{D_n(y_n) = 1\}| \geq \frac{1}{q(p(n))}$ . Значит  $y_n$  и  $G'(s)$  - не являются вычислительно неотличимыми. Значит  $G'(s)$  - не является ГПСЧ.

$$2. G'(s) = \begin{cases} 0^{|G(s)|}, & \text{если } s \text{ содержит ровно } \frac{|s|}{3} \text{ единиц} \\ G(s), & \text{иначе} \end{cases} \quad G'(s) - \text{не является ГПСЧ.}$$

В  $s$  ровно  $\frac{|s|}{3}$  единиц в  $C_n^{\frac{n}{3}}$  различных  $s$ . Воспользуемся формулой Стирлинга:

$$C_n^{\frac{n}{3}} = \frac{n!}{\frac{n}{3}! \frac{2n}{3}!} \sim \frac{3}{\sqrt{4\pi n}} \frac{3^n}{2^{\frac{2n}{3}}} \quad (3)$$

Обозначим событие  $X = \{s \text{ в } s \text{ ровно } \frac{|s|}{3} \text{ единиц}\}$ . Тогда, считая  $s \sim U_n$ , получим :

$$P\{G'(s) \neq G(s)\} \leq P\{X\} \sim \frac{3}{\sqrt{4\pi n}} \frac{3^n}{2^{\frac{2n}{3}}} \quad (4)$$

$\frac{3^n}{2^{\frac{2n}{3}}} = e^{n(\ln 3 - \frac{2}{3} \ln 2)}$ . Заметим, что  $\ln 3 - \frac{2}{3} \ln 2 = c < 0$ . Т.е.  $P\{G'(s) \neq G(s)\} \sim \frac{3}{\sqrt{\pi n}} e^{cn}$ . Значит  $\exists N \forall n > N : P\{G'(s) \neq G(s)\} \leq \frac{3}{\sqrt{\pi n}} e^{c_0 n} \leq e^{cn}$ ,  $c_0, c < 0$ .

Так как  $G(s)$  - ГПСЧ, то  $y_n \sim U_{p(n)}, \forall \{D_n\} \forall q_1(x)$  - полином  $\exists N \forall n > N :$   
 $|P\{D_n(G(s)) = 1\} - P\{D_n(y_n) = 1\}| < \frac{1}{q_1(p(n))}$ .

Воспользуемся определением вычислительной неотличимости :

$y_n \sim U_{p(n)}, \forall \{D_n\} \forall q(x)$  - полином  $\exists q_1(x) = \frac{q(x)}{2}, \exists N \forall n > N : |P\{D_n(G'(s)) = 1\} - P\{D_n(y_n) = 1\}| \leq |P\{D_n(G'(s)) = 1\} - P\{D_n(G(s)) = 1\}| + |P\{D_n(G(s)) = 1\} - P\{D_n(y_n) = 1\}| < e^{cn} + \frac{1}{q_1(p(n))} < \frac{1}{q(p(n))}$

Получили, что  $G'(s)$  и  $y_n$  вычислительно неотличимы. Значит  $G'(s)$  - ГПСЧ.

**Задача. 7.** Обобщённым sudoku называется такая задача: в квадрате  $n^2 \times n^2$  в некоторых клетках расставлены числа от 1 до  $n^2$ . Вопрос: можно ли заполнить оставшиеся клетки числами от 1 до  $n^2$ , так чтобы в каждой строке, в каждом столбце, а также в каждом из  $n^2$  "выровненных" квадратов  $n \times n$  каждое число встречалось по одному разу. В стандартном sudoku  $n = 3$ . Известно, что эта задача NP-полна. Предложите протокол доказательства существования решения с вычислительно нулевым разглашением, не использующий сводимость к какой-либо другой задаче.

**Ответ.** Будем использовать "Протокол привязки к биту", описанный на лекции, для выполнения операций "Загораживание" и "Открытие".

- $S$  - "загораживание",  $S(b) = (c, k)$
- $R$  - "открытие",  $R(c, k) = \{b, \text{ERROR}\}$
- Требования :
  1. Корректность :  $R(S(b)) = b$
  2. Секретность : привязка к 0 и 1 вычислительно неотличимы.
  3. Неподменяемость : невозможность  $R(c, k_0) = 0$  и  $R(c, k_1) = 1$

Протокол:

Исходная таблица  $T_0$ .

- $P$  выбирает случайную перестановку(перенумерацию)  $\sigma \in S_{n^2}$ , записывает решение в таблицу  $T_1$ , применяет  $\sigma$  к числам  $\{1, \dots, n^2\}$  из таблицы  $T_1$ . "Закрывает" таблицу  $T_1$  и перестановку  $\sigma$  и отправляет верификатору.
- $V$  выбирает случайным образом строку, столбец или квадрат, и просит прuverа "открыть" выбранный элемент в  $T_1$ . Также  $V$  выбирает случайную позицию  $(i, j)$  в исходной таблице такую, что  $T_0[i, j] = x$  (т.е. в  $T_0[i, j]$  уже записано некоторое известное число  $x$ , и просит прuverа "открыть"  $\sigma(x)$ . Проверяет на корректность строку, столбец или квадрат соответственно, а также проверяет, что  $T_1[i, j] = \sigma(x)$ .
- Повторяем протокол с начала нужное количество раз(полином).
- Принимает доказательство, если все проверки пройдены на каждом шаге.

Корректность:

- Если решение существует, то  $P$  будет действовать оптимально, ошибка второго рода может возникнуть только в протоколе привязки к биту, но ее можно сделать очень маленькой. Во всех остальных частях протокола ошибки не возникнет.
- Если решения нет, то либо есть некорректный элемент(строка, столбец, квадрат), либо прuver применил замену индексов не соответствующим перестановке  $\sigma$  образом. В первом случае вероятность не заметить ошибку  $P_1 \leq \frac{3n^2-1}{3n^2} = 1 - \frac{1}{3n^2}$ , во втором,  $P_2 \leq \frac{n^2-1}{n^2} = 1 - \frac{1}{n^2}$ . В любом случае  $P \leq 1 - \frac{1}{3n^2}$ .  $P^{3n^2} \leq (1 - \frac{1}{3n^2})^{3n^2} \sim \frac{1}{e}$ ,  $n \rightarrow \infty$ . Значит повторив полиномиальное количество раз сможем получить достаточно малую ошибку.

**Задача. 9.** Расширим определение РСР, введённое на лекции. Назовём классом  $\text{РСР}_{c,s}(r, q)_\Sigma$  класс языков  $L$ , для которых существует полиномиальный вероятностный верификатор  $V$  с произвольным доступом к строке  $\pi \in \Sigma^*$  длины не более  $q2^{O(R)}$ , со следующими условиями:

- $V$  использует не больше  $r$  случайных битов и делает не больше  $q$  неадаптивных запросов к  $\pi$  (обратите внимание, что здесь мы отказываемся от  $O(\cdot)$ -обозначений);
- Если  $x \in L$ , то  $\Pr\{V^\pi(x) = 1\} \geq c$ ;
- Если  $x \notin L$ , то  $\Pr\{V^\pi(x) = 1\} \leq s$ .

Соответственно, класс, введённый на лекции, является классом  $\mathbf{PCP}_{1, \frac{1}{2}}(r, q)_{\{0,1\}}$ . Будем считать, что размер алфавита  $|\Sigma|$  может зависеть от длины входа  $|x|$ . Докажите, что:

(а) (1 балл) Для алфавитов  $\Sigma$  полиномиального размера выполнено  $\mathbf{PCP}_{c,s}(O(\log n), 0)_{\Sigma} \subset \mathbf{P}$

(б) (4 балла) Для алфавитов  $\Sigma$  полиномиального размера выполнено  $\mathbf{PCP}_{c,s}(O(\log n), 1)_{\Sigma} \subset \mathbf{P}$  (для любых параметров  $c > s$ )

(в) (5 баллов) Для алфавитов  $\Sigma$  и параметров  $q$ , таких что  $|\Sigma|^q$  не больше полинома, выполнено  $\mathbf{PCP}_{1, \frac{1}{|\Sigma|q}}(O(\log n), q)_{\Sigma} \subset \mathbf{P}$

**Ответ.** 1.  $q = 0 \implies$  алгоритм проверки детерминированный, к сертификату не обращается, работает полином. Это есть  $\mathbf{P}$