Лекция 7

Доказательства с нулевым разглашением для *NP*.

(Конспект: И. Монахов)

Напоминание 7.1. Для пары алгоритмов V и P, обменивающихся сообщениями по протоколу $c(V,P)$, таких что алгоритм P (решатель) произвольный, а V (верификатор) полиномиальный вероятностный, обозначим через $\langle V,P\rangle(x)$ случайную величину, которая является последним сообщением от верификатора V в последовательсности сообщений $view(c(V(x),P(x)))$, где x – общая строчка входа V и P. Назовём (V,P) интерактивной системой доказательств для языка L, если

1. (V,P) полная: верификатор V принимает с ограниченной вероятностью ошибки доказательство решателя P для любой теоремы $x \in L$

$$\forall x \in L \quad \Pr\{\langle V,P\rangle(x) = 1\} \geq \frac{3}{4}.$$

2. (V,P) корректная: верификатор V отвергает с ограниченной вероятностью ошибки доказательство решателей P^* для всех строчек не из L

$$\forall x \notin L \quad \Pr\{\langle V,P^*\rangle(x) = 0\} \geq \frac{3}{4}.$$

Удобно рассматривать более "слабое" (на самом деле эквивалентное) определение интерактивного протокола.

Определение 7.1. Для пары алгоритмов V и P, обменивающихся сообщениями по протоколу $c(V,P)$, таких что алгоритм P (решатель) произвольный, а V (верификатор) полиномиальный вероятностный, обозначим через $\langle V,P\rangle(x)$ случайную величину, которая является последним сообщением от верификатора V в последовательсности сообщений $view(c(V(x),P(x)))$, где x – общая строчка входа V и P. Назовём (V,P) *интерактивной системой доказательств* для языка L, если

1. (V, P) полная: верификатор V принимает с ограниченной вероятностью ошибки доказательство решателя P для любой теоремы $x \in L$

$$\forall x \in L \quad \Pr\{\langle V, P\rangle(x) = 1\} = 1.$$

2. (V, P) корректная: верификатор V отвергает с ограниченной вероятностью ошибки доказательство решателей P^* для всех строчек не из L

$$\forall x \notin L \quad \Pr\{\langle V, P^*\rangle(x) = 0\} \geq \frac{1}{poly}.$$

Обозначим через G3C язык всех графов, которые можно расскрасить в три цвета так, чтобы никакое ребро не соединяло две вершины одного цвета. Этот язык задаёт **NP**-полную задачу поиска.

План доказательства: пусть язык $L \in \mathbf{NP}$. Рассмотрим сведение f, такое что $\forall x \quad x \in L \Leftrightarrow f(x) \in$ G3C. Чтобы получить интерактивное доказательство с нулевым разглашением для L можно вычислить функцию f, а потом использовать интерактивное доказательство с нулевым разглашением для G3C.

В этом плане есть следующие проблемы:

1. Даём $f(x)$, а у верификатора есть x, поэтому он может больше узнать.

2. Нужно повторить протокол, чтобы уменьшить вероятность ошибки.

Напоминание 7.2. Будем называть интерактивную систему доказательств (V, P) для языка L системой доказательств с вычисительно нулевым разглашением, если для любого полиномиального вероятностного верификатора V^* существует полиномиальный вероятностный симулятор M^*, такой что $\forall x \in L$ случайные величины $\langle V^*, P\rangle(x)$ и $M^*(x)$ вычислительно неразличимы.

Класс языков, для которых существует системы доказательств с вычислительно нулевым разглашением, будем обозначать **CZK**.

Определение 7.2. Рассмотрим для строчки $x \in L$ следующее обозначение: $P_L(x)$ – некоторое произвольное множество. Например, для графа $x \in$ G3C в качестве $P_L(x)$ можно взять одну из правильных раскрасок графа. Будем называть интерактивную систему доказательств (V, P) для языка L системой доказательств с *вычисительно нулевым разглашением относительно дополнительного входа*, если для любого полиномиального вероятностного верификатора V^* существует полиномиальный вероятностный симулятор M^*, такой что $\forall x \in L\ \forall y\ \forall z \in P_L(x)$ случайные величины $\langle V^*(y), P(z)\rangle(x)$ и $M^*(x, y)$ вычислительно неразличимы.

Введение дополнитиельного входа даёт возможно рассматривать более простые решатели – ведь им не нужно находить само доказательство.

Обозначение 7.1. Для решателя P с входом z и полинома q будем обозначать через $P^{q(|z|)}$ алгоритм, который $q(|z|)$ раз в цикле моделирует алгоритм $P(z)$.

Лемма 7.1. *Если* $\forall V^* \; \exists M^*$

$$\forall x \in L \; \forall y \; \forall z \in P_L(x) \;\; M^*(x,y) \text{ вычислительно неотличима от } \langle V^*(y), P(z)\rangle(x),$$

тогда $\forall V^* \; \exists M^{**}$

$$\forall x \in L \; \forall y \; \forall z \in P_L(x) \;\; M^{**}(x,y) \text{ вычислительно неотличима от } \langle V^{**}(y), P^{q(|x|)}(z)\rangle(x).$$

Доказательство. Работа V^{**} тоже разбивается на стадии (по сообщениям, посланным P^i). Построим новый алгоритм: когда закончилась стадия, со всех рабочих лент записываем информацию на дополнительный вход. Вначале следующей стадии переписываем обратно. Получим V^{***}, работающий по стадиям.

Есть симулятор на каждой стадии, применим гибридный метод: если различимы первая и последняя случайные величины, тогда различимы какие-то соседние – противоречие. □

Теорема 7.1. *Если существует протокол привязки к биту, тогда* `G3C` $\in$ **CZK**.

Доказательство. Пусть π – это алгоритм, упаковывающий бит в сундучок.

- Общий вход решателя P, верификаторов V, V^* и симулятора M^*: конечный граф $G = (V, E)$.

1. - V: пропускает свою очередь, посылая сообщение 0.
   - P: для правильной 3-раскраски $C \in P_L(x)$ отправляет сундучки $\pi(C(v_1)), \ldots, \pi(C(v_n))$.
2. - V: выбирает случайной ребро $(v_i, v_j) \in E$ и просит ключи от сундучков $\pi(C(v_i)), \pi(C(v_j))$.
   - P: отправляет K_i и K_j.
3. - V: если вершины v_i и v_j были правильно покрашены: $C(v_i) \neq C(v_j)$, посылает 1, посылает 1,
     иначе отправляет 0.
   - P: завершает протокол, отправив $\emptyset$.
4. - V: если нечестный решатель P^* не прислал пустое сообщение, завершает протокол, отправив $\emptyset$.
   - P^*: нет ограничений.

Опишем симулятор M^*.

1. - M_1: пропускает свою очередь, посылая сообщение 0.
   - M_2: генерирует случайную последовательность $c \in \{1,2,3\}^n$ и передаёт по протоколу привязки к биту $\pi(c_1), \pi(c_2), \ldots, \pi(c_n)$.

2.
   - M_1: выбирает случайной ребро $(v_i, v_j) \in E$ и просит ключи от сундучков $\pi(C(v_i)), \pi(C(v_j))$.
   - M_2: если повезло и $c_i \neq c_j$, отправляет K_i и K_j,
     иначе выдаёт $\perp$ и останавливается.
3.
   - M_1: если вершины v_i и v_j были правильно покрашены: $C(v_i) \neq C(v_j)$, посылает 1,
     иначе отправляет 0.
   - M_2: завершает протокол, отправив $\emptyset$.
4.
   - M_1: если нечестный решатель P^* не прислал пустое сообщение,
     завершает протокол, отправив $\emptyset$.
   - M_2^*: нет ограничений.

□

Замечание 7.1. Вместо $\perp$ можно всё стирать и повторять n раз.