Домашняя работа №11

Бредихин Александр

18 мая 2020 г.

Задача 1

Задача: Придумайте 0-knowledge протокол для доказательства того, что Мерлин знает решение некоторого судоку $n^2 \times n^2$.
Правила такие: доска $n^2 \times n^2$ разбивается на непересекающиеся блоки $n \times n$. В каждой ячейке может быть число от 1 до n^2. Изначально доска заполнена некоторым количеством чисел, которые в процессе заполнения менять нельзя. В каждом столбце, строке и блоке все числа должны быть разные.

Аналогично алгоритму из семинара: прувер случайным образом переставляет числа в правильном ответе, шифрует это а затем отвечает на запросы верефикатора (которому передаёт, что у него получилось). Верефикатор делает следующие забросы: либо раскрывает случайную строчку, либо случайный столбец, либо случайный выделенный квадрат (размером $n \times n$). Прувер даёт ему это и он проверяет это на выполнение правил судоку. Это последовательность действий повторяется сколько нужно раз (k).

Это алгоритм с нулевым разглашением, так как вся информация о правильном судоку – закодирована и после каждого запроса верефикатора происходит перекодирование. Найдём вероятность, что прувер не знает решение судоку, но верефикатор не смог определить это. Для этого найдём вероятность, что есть два повторяющихся числа в одном столбце, строке или квадратике и верефикатор обнаружит их. Её можно оценить $\frac{1}{n^2}$, так как нужно проверить 1 строчку, столбец или квадратик, где есть ошибка (то есть всего вариантов n^2 и подходит только один(оценка снизу)). Тогда:

$$P(ACC|G \notin \text{СУДОКУ}) \leq \left(1 - \frac{1}{n^2}\right)^k$$

Зная n можем сделать точность, которая нам нужна.

Задача 2

Задача: Докажите, что $\mathbf{AM} = BP \cdot \mathcal{NP}$.

Если у AM будет разреша двусторонняя ошибка, то случай становится тривиальным, так как для хода верефикатора (Артура) (который состоит из отправки случайных битов) (ему соответствует BP), прувер (Мэрлин) даст ответ который примет верефикатор. Эти два хода (ход верефикатора и прувера) соответствуют оценки NP.
Также нужно показать, что для $BP \cdot \mathcal{NP}$ можно будет свести к односторонней ошибки. Это (как я прочитал в интернете) можно сделать с помощью метода унивирсального хэширования коэффициента принятия верефикатора, так как он определяет высокую степень принятия с вероятностью 1.

Задача 3

Задача: Докажите, что $\#3SAT_D \in IP$. Для этого:

- Превратите булевы формулы в многочлены, значение которых совпадает с булевой формулой на одинаковом наборе. Такая операция называется арифметизацией.
- Определите, как с помощью арифметизации получить число выполняющих наборов.
- Постройте интерактивное доказательство того, что число выполняющих наборов действительное такое. Для этого понадобятся вычисления по модулю p.
- Оцените вероятность принятия для верификатора.

Построим по формуле $f = 3SAT$ многочлен P_f над некоторым полем F_p. Тогда f принадлежит $\#3SAT_D$ тогда и только тогда, когда для k, $bi \in \{0,1\}$ выполнено $P_f(b1, ..., bn) = 1$. Поскольку по построению значения многочлена равны либо нулю, либо единице, при $p > k$ это эквивалентно условию на сумму:

$$\sum_{(b_1,\ldots,b_n)\in\{0,1\}^n} P_f\left(b_1, b_2, \ldots, b_n\right) = k$$

Представим, как:

$$\sum_{b_1\in\{0,1\}}\sum_{b_2\in\{0,1\}}\cdots\sum_{b_n\in\{0,1\}}P_f\left(b_1,b_2,\ldots,b_n\right)=k$$

Раскрываем сумму:

$$\sum_{b_2\in\{0,1\}}\cdots\sum_{b_n\in\{0,1\}}P_f\left(0,b_2,\ldots,b_n\right)+\sum_{b_2\in\{0,1\}}\cdots\sum_{b_n\in\{0,1\}}P_f\left(1,b_2,\ldots,b_n\right)=k$$

Получаем: $A_1(0)+A_1(1)=k$, где

$$A_1(x)=\sum_{b_2\in\{0,1\}}\cdots\sum_{b_n\in\{0,1\}}P_f\left(x,b_2,\ldots,b_n\right)$$

Заметим, что A_1 является многочленом степени не больше, чем m. Эти коэффициенты мы передаём верефикатору в виде многочлена A_1'. Верификатор, получив многочлен A_1', проверяет $A_1'(0)+A_1'(1)=k$ (что число выполняющих наборов не изменяется) и говорит, что доказательство неверно, если проверка не прошла. Воспользуемся свойствами поля и малой степенью многочлена: если $A_1'\neq A_1$, то число таких а, что $A_1'(a)=A_1(a)$ не превосходит m. Верефикатор выбирает случайно $a_1\in F_p$ и даёт пруверу доказывать:

$$\sum_{b_2\in\{0,1\}}\cdots\sum_{b_n\in\{0,1\}}P_\varphi\left(a_1,b_2,\ldots,b_n\right)=A_1'\left(a_1\right)$$

Получили выражение, которое имеет тот же вид, но на одну переменную меньше, а в правой части стоит A_1' вместо k. Делаем аналогично и верефикатор ждёт от прувера коэффициенты многочлена

$$A_2(x)=\sum_{b_3\in\{0,1\}}\cdots\sum_{b_n\in\{0,1\}}P_f\left(a_1,x,b_3,\ldots,b_n\right)$$

прувер высылает многочлен A_2' и верефикатор снова проверяет условие $A_2'(0)+A_2'(1)=A_1(a_1)$ затем случайно выбирает a_2 и снова просит прувера доказать...

Так продолжается по индукции, пока не будут определены все $a1,a2,...,an$. Верификатору останется проверить, что

$$P_f\left(a_1,a_2,\ldots,a_n\right)=A_n'\left(a_n\right)$$

Он это делает самостоятельно (просто считает все скобки в получившимся многочлене).

Корректность протокола:
Если изначальное условие выполнено, то пруверу можно присылать $A'_i = A_i$, и все проверки пройдут с вероятностью 1. Если изначальное условие не выполнено, то в первом ходе у прувера есть два варианта. Если он присылает $A'_1 = A_1$, то проверка $A'_1(0) + A'_1(1) = k$ не проходит, и доказательство не принимается. Если же он присылает $A'_1 \neq A_1$, для которого проверка пройдёт, то с вероятностью не меньше $1 - \frac{m}{p}$ верификатор выберет такое a_1, что $A'_1(a_1) \neq A_1(a_1)$, и прувер останется с неправильным утверждением на втором ходе. Продолжая по индукции, получим, что с вероятностью не меньше $\left(1 - \frac{m}{p}\right)^n$ утверждение останется неверным и после n ходов, и тогда его ошибочность верификатор обнаружит при последней проверке. По неравенству Бернулли $\left(1 - \frac{m}{p}\right)^n > 1 - \frac{mn}{p}$, поэтому при $p > 3mn$ вероятность того, что неверное доказательство будет принято, будет меньше трети, что подходит в определении IP.

p находим так, что прувер присылает его, а верефикатор проверяет простоту (получено за полиномиальное время).