Лекция 3

Генераторы псевдослучайных чисел

(Конспект: С. Капулкин)

3.1 Генераторы псевдослучайных чисел

Определение 3.1. Пусть α_n, β_n — случайные величины со значениями в $\{0,1\}^{p(n)}$, где $p(n)$ — полином. Тогда α_n, β_n называются *вычислительно неразличимыми*, если для любого полиномиального вероятностного алгоритма A последовательность $\Pr[A(\alpha_n) = 1] - \Pr[A(\beta_n) = 1]$ пренебрежимо мала.

Утверждение 3.1. *Свойства вычислительной неразличимости:*

1. *Отношение* быть вычислительно неразличимым*: рефлексивно, симметрично и транзитивно.*

2. *Пусть α_n и β_n неразличимы, $q(n)$ – полином. Тогда $\alpha_n U_{q(n)}$ и $\beta_n U_{q(n)}$ вычислительно неразличимы, где $U_{q(n)}$ — это случайная величина, распределенная равномерно на строчках длины $q(n)$.*

3. *Пусть α_n и β_n неразличимы, f – полиномиально вычислимая функция $\Longrightarrow$ $f(\alpha_n)$ и $f(\beta_n)$ вычислительно неразличимы.*

Доказательство. Утверждение 1 очевидно. Утверждения 2 и 3 доказываются аналогично. Докажем, например, третье. Пусть полиномиальный вероятностный алгоритм A различает $f(\alpha_n)$ и $f(\beta_n)$. Тогда алгоритм, который на входе x выдает $A(f(x))$ различает α_n и β_n. □

Определение 3.2. Пусть $G : \{0,1\}^n \to \{0,1\}^{l(n)}$ — полиномиально вычислимая функция, где $l(n) > n$. $G(n)$ называется $l(n)$-генератором псевдослучайных чисел, если $G(U_n)$ вычислительно неотличимо от $U_{l(n)}$.

Теорема 3.1. *Если $G(n)$ — это $2n$-генератор псевдослучайных чисел, то G — сильно односторонняя функция.*

Доказательство. Пусть G не является сильно односторонней функцией, тогда существует такой вероятностный полиномиальный по времени алгоритм A и полином $p(n)$, что $\Pr_{x\in U_n}[A(G(x)) \in G^{-1}(G(x))] \geq \frac{1}{p(n)}$ для бесконечного числа n. Построим алгоритм A', который отличит $G(U_n)$ от U_{2n}.

Алгоритм A' получает на вход $y \in \{0,1\}^{2n}$:

- Вычисляет $z = A(y)$
- Если $G(z) = y$, то выдать 1, иначе 0.

Для бесконечного числа n выполняется $\Pr_{y\in G(U_n)}[A'(y) = 1] \geq 1/p(n)$. Для всех n выполняется $Pr_{y\in U_{2n}}[A'(y) = 1] \leq \frac{2^n}{2^{2m}} = 1/2^n$ Получаем противоречие с тем, что G — это генератор псевдослучайных чисел. □

На самом деле верна и обратная теорема:

Теорема 3.2 (Голдрейх, Импальяццио, Луби, Хастад). *Если существуют односторонние функции, то существует $p(n)$-генератор псевдослучайных чисел для любого полинома $p(n)$.*

Мы докажем только частный случай этой теоремы в более сильном предположении, если существуют односторонние перестановки. Для начала докажем, что существует генератор псевдослучайных чисел, который увеличивает вход всего на 1 бит.

Лемма 3.1. *Если существует биективная сохраняющая длину односторонняя функция (такие функции мы будем называть перестановками), то существует $(n+1)$-генератор псевдослучайных чисел.*

Доказательство. Построим псевдослучайный генератор конструктивно. Пусть f — односторонняя перестановка, B — это ее трудный бит. Покажем, что $G(x) = f(x)B(x)$ — это $(n+1)$-генератор.

Докажем это от противного, пусть G не является генератором псевдослучайных чисел, алгоритм A различает $G(U_n)$ и U_{n+1} с вероятностью $\geq 1/p(n)$, где $p(n)$ — это некоторый полином, т.е. для бесконечного числа n выполняется:

$$|\Pr_{x\leftarrow U_n}[A(f(x)B(x))=1] - \Pr_{y\leftarrow U_n, b\in U_1}[A(yb)=1]| \geq 1/p(n) \tag{3.1}$$

Введем обозначения: $Pr_{x\leftarrow U_n}[A(f(x)B(x))=1] = \alpha$ и $Pr_{x\leftarrow U_n}[A(f(x)\bar{B(x)})=1] = \beta$.

Пользуясь тем, что f — перестановка, можно записать:

$$\begin{aligned}\Pr[A(yb)=1] = \Pr[A(f(y)b)=1] = \Pr[A(f(y)B(y))=1]Pr[b=B(y)] \\ + \Pr[A(f(y)\bar{B(y)})=1]Pr[b=\bar{B(y)}] = (\alpha+\beta)/2\end{aligned}$$

Из неравенства (3.1) следует, что $|\alpha - \frac{\alpha+\beta}{2}| \geq 1/p(n)$, т.е. $|\frac{\alpha-\beta}{2}| \geq 1/p(n)$.

Не умаляя общности, будем считать, что для бесконечного числа n выполняется неравенство $\alpha > \beta$ (в противном случае можно поменять ответ алгоритма A на противоположный). Построим алгоритм C, который будет вычислять трудный бит.

Алгоритм C получает на вход $y \in \{0,1\}^n$:

- выбирает $b \in U_1$
- если $A(yb) = 1$, то выдает b, иначе $1 - b$.

Проверим, что алгоритм C, действительно, вычисляет трудный бит с нужной вероятностью (мы опять воспользуемся, что f — перестановка):

$$\begin{aligned}
&\Pr[C(y)=B(x)] \\
&\quad = \Pr[C(y)=B(x) \mid b=B(x)]Pr[b=B(x)] + \Pr[C(y)=B(x) \mid b\neq B(x)]\Pr[b\neq B(x)] \\
&\qquad = \frac{1}{2}\Pr[A(f(x)B(x)) = 1] + \frac{1}{2}\Pr[A(f(x)\bar{B(x)}) \neq 1] \\
&\qquad\qquad = \frac{1}{2}\alpha + \frac{1}{2}(1-\beta) = \frac{1}{2} + \frac{\alpha-\beta}{2} \geq \frac{1}{2} + \frac{1}{p(n)}.
\end{aligned}$$

□

Теорема 3.3. *Пусть f - односторонняя перестановка, B - трудный бит для нее, $p(n)$ — это полином. Тогда $B(x)B(f(x))\dots B(f^{p(n)-n-1}(x))f^{(p(n)-n)}(x)$ является $p(n)$-генератором псевдослучайных чисел.*

Доказательство. Пусть $B(x)B(f(x))\dots B(f^{p(n)-n-1}(x))f^{(p(n)-n)}(x)$ отличима от $U_{p(n)}$ и алгоритм A отличает распределения. Тогда построим алгоритм, отличающий $B(x)f(x)$ от $bf(x)$, где $x \leftarrow U_n$, а $b \leftarrow U_1$, тем самым получим противоречие с предыдущей леммой.

Для доказательства мы используем гибридный метод. Алгоритм A отличает два распределения друг от друга, мы построим цепочку распределений полиномиальной длины, которые плавно переходят от одного распределения в другое. Рассмотрим цепочку распределений $D_0, D_2, \dots, D_{p(n)-n}$, где $D_{p(n)-n} = U_{p(n)}$, а $D_0 = B(x)B(f(x))\dots B(f^{p(n)-n-1}(x))f^{(p(n)-n)}(x)$.

Пусть $b_1, b_2, \dots, b_{p(n)-n-1} \leftarrow U_1, y \leftarrow U_n$. Определим промежуточные распределения так:

$$\begin{array}{lllll}
B(x) & B(f(x)) & \dots & B(f^{(p(n)-n-1)}(x)) & f^{(p(n)-n)}(x) \quad : D_0 \\
b_1 & B(x) & \dots & & f^{(p(n)-n-1)}(x) \quad : D_1 \\
b_1 & b_2 & B(x) & \dots & f^{(p(n)-n-2)}(x) \\
\vdots & & & & \\
b_1 & b_2 & & B(x) & f(x) \quad : D_{p(n)-n-1} \\
b_1 & b_2 & \dots & b_{p(n)-n-1} & y \quad : D_{p(n)-n}
\end{array}$$

Если полиномиальный вероятностный алгоритм A различает D_0 и $D_{p(n)-n}$ (для бесконечного числа длин входов n):

$$|\Pr_{x\leftarrow D_0}[A(x)=1] - \Pr_{x\in D_{p(n)-n}}[A(x)=1]| \geq 1/q(n),$$

где $q(n)$ — это полином, то найдется такое $0 \leq i \leq p(n)-n-1$, что алгоритм A различает D_i и D_{i+1}:

$$|\Pr_{x\leftarrow D_i}[A(x)=1] - \Pr_{x\leftarrow D_{i+1}}[A(x)=1]| \geq \frac{1}{(p(n)-n)q(n)}.$$

Заметим, что $i < p(n) - n - 1$, так как $D_{p(n)-1}$ и $D_{p(n)}$ вычислительно неотличимы по предыдущей лемме.

Рассмотрим распределения $D_i = b_1 b_2 \dots b_i B(x) B(f(x)) \dots B(f^{p(n)-n-i-1}(x)) f^{p(n)-n-i}(x)$ и $D_{i+1} = b_1 b_2 \dots b_i B(x) B(f(x)) \dots B(f^{p(n)-n-i-1}(x)) f^{p(n)-n-i}(x)$.

D_i и D_{i+1} отличимы алгоритмом A, следовательно отличимы распределения $H_1 = b_i B(x) B(f(x)) \dots$ и $H_2 = B(x) B(f(x)) \dots$. Поскольку f — это перестановка, то H_1 неотличимо от $H_3 = b_i B(f(x)) B(f(f(x))) \dots$, т.е. H_2 и H_3 вычислительно отличимы. Тогда вычислительно отличимы распределения $b_i f(x)$ и $B(x) f(x)$, а это противоречит предыдущей лемме.

Осталось найти значение i, для которого алгоритм A отличает D_i от D_{i+1} с достаточно хорошей вероятностью. Для этого достаточно промоделировать все распределения $D_0 \dots D_{p(n)-n}$. На каждом распределении D_j следует запустить алгоритм A большое количество раз (достаточно $(p(n)q(n))^5$) и получить вероятность того, что алгоритм A выдаст 1 на этих распределениях приближенно. Из оценок Чернова следует, что с вероятностью $1 - 2^{-n}$ ошибка будет не более, чем на $\frac{1}{(p(n)q(n))^2}$, а этого достаточно, чтобы найти i. В следующей лекции мы рассмотрим способ, как в гибридном методе можно избежать поиска i (грубо говоря, нужное i можно угадать).

□

3.2 Протоколы с секретным ключом

Задача шифрования с секретным ключом состоит в том, что двое участников договариваются об общей секретной строке (секретном ключе), такую строку лучше всего сгенерировать случайным образом. После чего один из участников может послать зашифрованное сообщение второму участнику, прочитать посланное сообщение может только тот, кто знает секретный ключ. Решит такую задачу можно, например, с помощью схемы *гамирования*, которая заключается в том, что сообщение побитово ксорится (складывается по модулю два) с ключом. Чтобы сообщение расшифровать его нужно снова поксорить с ключом. Недостатком такого шифрования является то, что размер секретного ключа должен быть равен размеру сообщения. Еще один недостаток состоит в том, что такую схему можно применять ровно один раз. Мы поборемся с первым недостатком схемы гамирования с помощью генераторов псевдослучайных чисел.

Определение 3.3. Протоколом с секретным ключом называется пара полиномиальных вероятностных алгоритма E и D. Алгоритм E называется шифратором, он получает на вход пару строчек (e, m), где e — это секретный ключ, а m — это сообщение, которое нужно зашифровать. Результат работы алгоритма E — это сторка (шифрограмма). Алгоритм D получает на вход ключ e и шифрограмму x и выдает сообщение m. Алгоритмы E и D обладают двумя свойствами:

1. Для всех e и m выполняется $\Pr[D(e, E(e, m)) = m] = 1$.

2. для любого полинома $q(n)$ для любых последовательностей строчек a_n и b_n: $|a_n| = |b_n| = q(n)$, $E(e, a_n)$ и $E(e, b_n)$ вычислительно неразличимы, где $e \leftarrow U_n$

(в случайных величинах $E(e, a_n)$ и $E(e, b_n)$ учитываются случайные биты алгоритма E).

Теорема 3.4. *Если существуют генераторы псевдослучайных чисел, то существует протокол с закрытым ключом. Пусть E получает на вход ключ e длины n и сообщение m длины $q(n)$. Пусть G — это $q(n)$-генератор псевдослучайных чисел. Определим $E(e, m) = G(e) \oplus m$, а $D(e, x) = x \oplus G(e)$. Так определенные E и D являются протоколом с закрытым ключом.*

Доказательство. Проверим, что выполняется второе свойство из определения 3.3 (выполнение первого свойства очевидно). Пусть $G(e) \oplus m_1$ вычислительно отличимо от $G(e) \oplus m_2$ для каких-то двух сообщений m_1 и m_2, тогда для какого-то $i \in \{1, 2\}$ (это i можно найти моделированием) случайная величина $G(e) \oplus m_i$ отличается от $G(e) \oplus 0^{q(n)}$, т.е. $G(e) \oplus m_i$ отличается от $G(e)$, поскольку $G(e)$ вычислительно неотличимо от $U_{q(n)}$, то $U_{q(n)} \oplus m_i$ отличается от $G(e)$, что означает, что $U_{q(n)}$ вычислительно отличимо от $G(e)$ при $e \leftarrow U_n$, а это противоречит с тем, что G — это $q(n)$-генератор псевдослучайных чисел. □