

„Теория информации“.

Лекции 1-9

А.В. Смаль

27 апреля 2017 г.

Комбинаторный подход

Информация по Хартли

Пусть задано некоторое конечное множество A — *множество исходов*.

Определение 1.1 (1928). Определим *количество информации* в A как $\chi(A) = \log_2 |A|$ (мы будем измерять количество информации в битах, поэтому все логарифмы будут по основанию 2, для байтов основание нужно было бы заменить на 256).

Если про некоторый $x \in A$ стало известно, что $x \in B$, то теперь для идентификации x нам достаточно $\chi(A \cap B) = \log |A \cap B|$ битов, т.е. нам сообщили $\chi(A) - \chi(A \cap B)$ битов информации.

Пример 1.1. Предположим, что мы хотим узнать некоторое неизвестное упорядочение множества $\{a_1, a_2, \dots, a_5\}$. Нам стало известно, что $a_1 > a_2$ или $a_3 > a_4$. Сколько битов информации мы узнали? Множество A состоит из $5!$ перестановок, множество B — из перестановок, которые удовлетворяют новому условию. Легко проверить, что $|B| = 90$. Итого мы узнали $\log 120 - \log 90 = \log(4/3)$ битов.

Пусть $A \subset \{0, 1\}^* \times \{0, 1\}^*$. Обозначим через $\pi_1(A)$ и $\pi_2(A)$ проекции множества A на первую и вторую координату соответственно, а $\chi_1(A) = \log |\pi_1(A)|$ и $\chi_2(A) = \log |\pi_2(A)|$ — их сложность по Хартли.

Теорема 1.1. $\chi(A) \leq \chi_1(A) + \chi_2(A)$.

Определение 1.2. Количество информации в второй компоненте A при известной первой

$$\chi_{2|1} = \log \left(\max_{a \in \pi_1(A)} |A_a| \right),$$

где $A_a = \{(a, x) \mid x \in \pi_2(A)\}$.

Теорема 1.2. $\chi(A) \leq \chi_1(A) + \chi_{2|1}(A)$.

Теорема 1.3. Для $A \subset \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^*$

$$2 \cdot \chi(A) \leq \chi_{12}(A) + \chi_{13}(A) + \chi_{23}(A).$$

Следствие 1.1. Квадрат объёма трёхмерного тела не превосходит произведение площадей его проекций на координатные плоскости.

Утверждение 1.1. Если $f : X \rightarrow Y$

1. является сюръекцией, то $\chi(Y) \leq \chi(X)$,
2. является инъекцией, то $\chi(X) \leq \chi(Y)$.

Применение: игра в 10 вопросов

Сколько вопросов на ДА/НЕТ нужно задать, чтобы определить загаданное число от 1 до N , если (а) можно задавать вопросы адаптивно; (б) вопросы нужно написать на бумажке заранее.

Оценка $\lceil \log N \rceil$ достигается в обоих случаях, если задавать вопросы про биты двоичного представления загаданного числа.

Докажем нижнюю оценку. Пусть $A = [N]$. Множество $Q = \{(q_1, q_2, \dots, q_k)\}$ — множество протоколов (ответы на вопросы). Можно рассматривать A и Q как проекции некоторого множества исходов игры S на разные координаты. Тогда верны следующие неравенства:

- $\chi_Q(S) = \chi(Q) \leq \chi_1(Q) + \chi_2(Q) + \dots + \chi_k(Q) \leq k$,
- $\chi_A(S) = \chi(A) \leq \chi(S) \leq \chi_Q(S) + \chi_{A|Q}(S) \leq k + 0 = k$.

Таким образом получаем, что $\log N = \chi(A) \leq k$.

Цена информации

Пусть имеется некоторое неизвестное число от 1 до n (где $n \geq 2$). Разрешается задавать любые вопросы с ответами ДА/НЕТ. При ответе ДА мы заплатим 1 рубль, а при ответе НЕТ — два рубля. Сколько необходимо и достаточно заплатить для отгадывания числа?

Верхняя оценка. Давайте задавать вопросы так, чтобы отрицательные ответы приносили в два раза больше информации, чем положительные. Тогда за каждый бит информации мы заплатим $c \log n$ для некоторой константы c . Пусть вопросы будут вида „ $x \in T$?“. Тогда требуется

$$2(\log |X| - \log |X \cap T|) = \log |X| - \log |X \cap \bar{T}|.$$

Пусть $|X \cap T| = \alpha |X|$, тогда $|X \cap \bar{T}| = (1 - \alpha)|X|$, т.о. $\alpha^2 = 1 - \alpha$, $\alpha = (\sqrt{5} - 1)/2$. При любом ответе мы заплатим $c = 1/(-\log \alpha) \approx 1.44$ рублей за бит, а в целом — $\log n/(-\log \alpha)$ рублей.

Нижняя оценка. Применим рассуждение про злонамеренного противника (adversary argument). Пусть противник выбирает ответ ДА/НЕТ в зависимости от того, какое из двух значение $1/(\log |X| - \log |X \cap T|)$ и $(2/\log |X| - \log |X \cap \bar{T}|)$ больше. При любых X, T одно из этих значений не меньше $c = 1/(-\log \alpha)$. Таким образом мы заставляем алгоритм платить не менее c рублей за бит, а значит любой алгоритм в худшем случае заплатит $\lceil c \log n \rceil$ рублей.

Применение: упорядочивание камней по весу

Верхняя и нижняя оценки для произвольного N

Сколько сравнений нужно сделать для того, чтобы упорядочить N камней по весу?

Нижняя оценка. Потребуется $\lceil \chi(S_N) \rceil = \lceil \log n! \rceil$ сравнений.

Верхняя оценка. Будем сортировать вставкой с бинарным поиском места вставки. Количество сравнений:

$$\lceil \log 2 \rceil + \lceil \log 3 \rceil + \dots + \lceil \log n \rceil \leq \log n! + n - 1 = n \log n + O(n).$$

Точные оценки для маленьких N

Упражнение 1.1. Сколько нужно взвешиваний, чтобы упорядочить N камней по весу? Найдите точный ответ на этот вопрос для $N = 2, 3, 4, 5$. Указание: воспользуйтесь жадной стратегией, при которой каждое взвешивание приносит максимум информации.

Применение: поиск фальшивой монетки

- 20 монет, одна фальшивая легче остальных.

Каждое взвешивание даёт не более $\log 3$ битов. Итого $k \geq \log N / \log 3 = \log_3 N$.

- 13 монет, одна фальшивая (с неизвестным относительным весом), 3 взвешивания.

Два варианта первого шага:

- если взвешиваем по 4, то при равенстве нельзя из 5 за два взвешивания найти фальшивую (остаётся 10 исходов),
- если взвешиваем по 5, то при неравенстве остаётся 10 возможных исходов.

- 15 монет, одна фальшивая, три взвешивания. Не требуется узнавать относительный вес монеты.

Всего исходов $2 \cdot 14 + 1 > 27$, т.к. только в случае трёх равенств мы можем не узнать относительный вес фальшивой монеты.

- 14 монет, одна фальшивая, три взвешивания. Не требуется узнавать относительный вес монеты.

Всего исходов $2 \cdot 13 + 1 \leq 27$, но определить тем не менее нельзя. Аппарата информации по Хартли недостаточно.

Логика знаний

В этом разделе мы будем называть множество исходов A множеством *миров*. Пусть f — это некоторая функция из A в некоторое множество I (будем воспринимать это как информация о мире). Нам не важно какие значения принимает f , нам будут важны лишь классы эквивалентности, на которые f разбивает A : каждый класс эквивалентности будет состоять из миров A с одинаковым значением f .

Пример 1.2. Пусть $A = \{1, 2, 3, 4, 5\}$, а $f(x) = x \bmod 3$. Тогда f разбивает A на три класса эквивалентности $\{1, 4\}$, $\{2, 5\}$ и $\{3\}$.

Пусть $B \subset A$ — это некоторое *утверждение* о мирах. B *истинно* в мире x , если $x \in B$. В противном случае B *ложно* в x . В мире x мы *знаем*, что B *истинно*, если $y \in B$ для всех $y \sim x$.

Пример 1.3. Пусть $A = \{1, 2, 3, 4, 5\}$, а $f(x) = x \bmod 3$. Тогда в мирах 1, 4 и 3 мы знаем, что мир меньше 5. А в мирах 2 и 5 — не знаем.

Замечание 1.1. „Не знаем“ мы будем понимать в смысле „не верно, что знаем“.

К утверждениям о мирах можно применять обычные логические связки: «И» (пересечение), «ИЛИ» (объединение), «НЕ» (дополнение).

Утверждение 1.2. *Если в мире x мы знаем B , то в мире x мы знаем, что мы знаем B . Аналогично, если в мире x мы не знаем B , то в мире x мы знаем, что не знаем B .*

Пусть теперь у нас есть k человек со своими знаниями о мире. Они определяют k отношений эквивалентности $\sim_1, \sim_2, \dots, \sim_k$ и, соответственно, k разбиений на классы эквивалентности.

Пример 1.4. Пусть множество миров $A = \{1, 2, 3, 4, 5\}$ и есть два человека, Алиса и Боб. Алиса знает значения $f_A(x) = x \bmod 3$, а Боб знает $f_B(x) = x \bmod 2$. Тогда классы эквивалентности Алисы: $\{1, 4\}$, $\{2, 5\}$ и $\{3\}$, а классы эквивалентности Боба: $\{1, 3, 5\}$ и $\{2, 4\}$. В мире 1 Алиса знает, что мир меньше 5, а Боб не знает. В мире 4 они оба это знают. В мире 1 Алиса не знает, что Боря не знает, что мир меньше 5 (действительно, в мире 4, который с точки зрения Алисы эквивалентен 1, Боря это знает).

Вероятностный подход

Энтропия Шеннона

Энтропия Шеннона определяет количество информации $H(\alpha)$ в распределении вероятностей для некоторой случайной величины α . Пусть α принимает значения из множества $\{a_1, a_2, \dots, a_k\}$ с вероятностями $\{p_1, p_2, \dots, p_k\}$, $p_i \geq 0$, $\sum_i p_i = 1$.

Нам бы хотелось, чтобы это определение согласовывалось с определением Хартли, т.е. имеют место следующие „граничные условия“:

- если $p_1 = \dots = p_k$, то $H(\alpha) = \log k$,
- если $p_1 = 1, p_2 = \dots = p_k = 0$, то $H(\alpha) = 0$.

Будем искать $H(\alpha)$ в виде математического ожидания „удивления“ от исхода случайной величины („удивление“ зависит от вероятности данного исхода).

$$H(\alpha) = \sum_i p_i \cdot \text{impress}(p_i).$$

Граничные условия однозначно определяют функцию $\text{impress}(p_i) = \log \frac{1}{p_i} = -\log p_i$.

Определение 2.1 (1948). Энтропия Шеннона случайной величины α

$$H(\alpha) = \sum_{i=1}^k p_i \cdot \log \frac{1}{p_i}.$$

(По непрерывности доопределим $0 \cdot \log \frac{1}{0} = 0$.)

Можно вывести это соотношение из определения Хартли более формально. Пусть W_n — это множество всех слов длины n состоящих из букв $\{a_1, a_2, \dots, a_k\}$, где каждая буква a_i встречается ровно $n_i = p_i \cdot n$ раз (будем считать, что вероятности p_i рациональны, и что множество W_n определено только тогда, когда все n_i целые). Информация по Хартли в W_n

$$\chi(W_n) = \log |W_n| = \log \frac{n!}{n_1! n_2! \dots n_k!}.$$

Это выражение можно оценить при помощи формулы Стирлинга.

$$\begin{aligned} \chi(W_n) &= \log \frac{\text{poly}(n) \cdot (n/e)^n}{\text{poly}(n) \cdot (n_1/e)^{n_1} \cdot (n_2/e)^{n_2} \dots (n_k/e)^{n_k}} = \\ &= \log \left(\left(\frac{n}{n_1} \right)^{n_1} \cdot \left(\frac{n}{n_2} \right)^{n_2} \dots \left(\frac{n}{n_k} \right)^{n_k} \right) + O(\log n) = \\ &= \log \left(\left(\frac{1}{p_1} \right)^{p_1 \cdot n} \cdot \left(\frac{1}{p_2} \right)^{p_2 \cdot n} \dots \left(\frac{1}{p_k} \right)^{p_k \cdot n} \right) + O(\log n) = \\ &= n \cdot \sum_{i=1}^k p_i \cdot \log \frac{1}{p_i} + O(\log n). \end{aligned}$$

В среднем на один символ приходится $\chi(W_n)/n$ битов информации. В пределе получаем

$$\lim_{n \rightarrow \infty} \frac{\chi(W_n)}{n} = \sum_{i=1}^k p_i \cdot \log \frac{1}{p_i} = H(\alpha)$$

(предел нужно брать по бесконечной подпоследовательности натуральных чисел n таких, для которых все $\{n_i\}$ — целые).

Лемма 2.1. Для энтропии Шеннона выполняются следующие соотношения.

- $H(\alpha) \geq 0$, причём $H(\alpha) = 0 \iff$ распределение α вырождено.
- $H(\alpha) \leq \log k$, причём $H(\alpha) = \log k \iff$ величина α распределена равномерно.

Для доказательства нам потребуется следующая теорема.

Теорема 2.1 (Неравенство Йенсена). Пусть функция $f(x)$ является вогнутой на некотором промежутке \mathcal{X} и числа $q_1, q_2, \dots, q_n > 0$ таковы, что $q_1 + \dots + q_n = 1$. Тогда для любых x_1, x_2, \dots, x_n из промежутка \mathcal{X} выполняется неравенство:

$$\sum_{i=1}^n q_i f(x_i) \leq f\left(\sum_{i=1}^n q_i x_i\right).$$

Доказательство леммы 2.1. Первое свойство следует напрямую из определения: каждый член суммы $H(\alpha)$ неотрицателен и равен нулю только в случае, если $p_i = 0$ или $p_i = 1$.

Для доказательства второго неравенства перенесём всё в левую часть и применим неравенство Йенсена:

$$H(\alpha) - \log k = \sum_{i=1}^k p_k \cdot \log \frac{1}{p_i} - \sum_{i=1}^k p_i \cdot \log k = \sum_{i=1}^k p_k \cdot \log \frac{1}{p_i k} \leq \log \left(\sum_{i=1}^k p_i \frac{1}{p_i k} \right) = \log 1 = 0.$$

□

Энтропию совместного распределения пары случайных величин α и β будем обозначать $H(\alpha, \beta)$.

Лемма 2.2. Выполняются следующие свойства:

- $H(\alpha, \beta) \leq H(\alpha) + H(\beta)$, причём равенство достигается тогда и только тогда, когда случайные величины независимы;
- $H(\alpha) \leq H(\alpha, \beta)$, причём равенство достигается тогда и только тогда, когда β полностью определяется значением α , т.е. $\beta = f(\alpha)$.

Доказательство. Введём обозначения для вероятностей событий совместного распределения вероятностей (α, β) . Пусть пара (a_i, b_j) имеет вероятность $p_{i,j}$, событие $[\alpha = a_i]$ имеет вероятность $p_{i,*} = p_{i,1} + \dots + p_{i,n}$, а событие $[\beta = b_j]$ — вероятность $p_{*,j} = p_{1,j} + \dots + p_{k,j}$. В этих обозначениях неравенство $H(\alpha, \beta) \leq H(\alpha) + H(\beta)$ переписывается как

$$\sum_{i,j} p_{i,j} \cdot \log \frac{1}{p_{i,j}} \leq \sum_i \sum_j p_{i,j} \cdot \log \frac{1}{p_{i,*}} + \sum_j \sum_i p_{i,j} \cdot \log \frac{1}{p_{*,j}}.$$

Перенесём всё в левую часть и применим неравенство Йенсена.

$$\begin{aligned} \sum_{i,j} p_{i,j} \cdot \log \frac{p_{i,*} \cdot p_{*,j}}{p_{i,j}} &\leq \log \left(\sum_{i,j} p_{i,j} \cdot \frac{p_{i,*} \cdot p_{*,j}}{p_{i,j}} \right) = \log \left(\sum_{i,j} p_{i,*} \cdot p_{*,j} \right) = \\ &= \log \left(\underbrace{\sum_i p_{i,*}}_1 \cdot \underbrace{\sum_j p_{*,j}}_1 \right) = 0. \end{aligned}$$

Равенство в неравенстве Йенсена для $f(x) = \log(x)$ достигается только, если все точки равны, т.е. для любых i, j $\frac{p_{i,*} p_{*,j}}{p_{i,j}} = c$ для некоторой константы c . Несложно заметить, что $c = 1$, т.к. выполняется следующее равенство $\sum_{i,j} p_{i,*} p_{*,j} = c \sum_{i,j} p_{i,j}$ в котором обе суммы равны 1. Таким образом в случае равенства α и β независимы.

Доказательство второго свойства мы получим как следствие из свойств условной энтропии. \square

Определение 2.2. Энтропия α при условии $\beta = b_j$

$$H(\alpha \mid \beta = b_j) = \sum_i \Pr[\alpha = a_i \mid \beta = b_j] \cdot \log \frac{1}{\Pr[\alpha = a_i \mid \beta = b_j]}.$$

Определение 2.3. Условная (относительная) энтропия α относительно β

$$H(\alpha \mid \beta) = \sum_j \Pr[\beta = b_j] \cdot H(\alpha \mid \beta = b_j).$$

Другими словами

$$H(\alpha \mid \beta) = \mathbb{E}_{b_j \leftarrow \beta} [H(\alpha \mid \beta = b_j)].$$

Если подставить определение 2.2, то можно получить выражение для условной энтропии через отдельные вероятности событий.

$$H(\alpha \mid \beta) = \sum_j \Pr[\beta = b_j] \cdot \sum_i \Pr[\alpha = a_i \mid \beta = b_j] \cdot \log \frac{1}{\Pr[\alpha = a_i \mid \beta = b_j]} = \sum_{i,j} p_{i,j} \cdot \log \frac{p_{*,j}}{p_{i,j}}.$$

Лемма 2.3. Условная энтропия обладает следующими свойствами.

- $H(\alpha \mid \beta) \geq 0$.
- $H(\alpha \mid \beta) = 0 \iff \alpha$ однозначно определяется по β .
- $H(\alpha, \beta) = H(\beta) + H(\alpha \mid \beta) = H(\alpha) + H(\beta \mid \alpha)$.

Доказательство. Первое свойство выполняется, т.к. условная энтропия это матожидание неотрицательной случайной величины. Второе свойство объясняется тем, что для любого j распределение $\langle \alpha \mid \beta = b_j \rangle$ имеет нулевую энтропию, т.е. распределение вырождено и каждому b_j соответствует ровно один a_i . Третье свойство следует из следующего равенства.

$$\sum_{i,j} p_{i,j} \cdot \log \frac{1}{p_{i,j}} = \sum_{i,j} p_{i,j} \cdot \log \frac{1}{p_{*,j}} + \sum_{i,j} p_{i,j} \cdot \log \frac{p_{*,j}}{p_{i,j}}.$$

(Нужна аккуратность, если есть строки, которые состоят из одних нулей, т.е. $p_{*,j} = 0$ — такие строки не нужно включать в эти суммы.) \square

Следствие 2.1. $H(\alpha, \beta) \geq H(\alpha)$, причём равенство достигается тогда и только тогда, когда $\beta = f(\alpha)$.

Доказательство. $H(\alpha, \beta) - H(\alpha) = H(\beta \mid \alpha) \geq 0$. По второму свойству условной энтропии равенство достигается тогда и только тогда, когда $\beta = f(\alpha)$. \square

Взаимная информация

Определение 2.4. *Информация в α о величине β* определяется следующим соотношением:

$$I(\alpha : \beta) = H(\beta) - H(\beta \mid \alpha).$$

Эту величину так же называют *взаимной информацией случайных величин α и β* .

Лемма 2.4. *Для взаимной информации выполняются следующие соотношения.*

1. $I(\alpha : \beta) \leq H(\alpha)$.
2. $I(\alpha : \beta) \leq H(\beta)$.
3. $I(\alpha : \alpha) = H(\alpha)$.
4. $I(\alpha : \beta) = I(\beta : \alpha)$.
5. $I(\alpha : \beta) = H(\alpha) + H(\beta) - H(\alpha, \beta)$.

Определение 2.5. Пусть α, β, γ — случайные величины. Определим *взаимную информацию в α о β при условии γ* .

1. $I(\alpha : \beta \mid \gamma) = H(\beta \mid \gamma) - H(\beta \mid \alpha, \gamma)$.
2. $I(\alpha : \beta \mid \gamma) = \sum_{\ell} I(\alpha : \beta \mid \gamma = c_{\ell}) \cdot \Pr[\gamma = c_{\ell}]$.
3. $I(\alpha : \beta \mid \gamma) = H(\alpha \mid \gamma) + H(\beta \mid \gamma) - H(\alpha, \beta \mid \gamma)$.
4. $I(\alpha : \beta \mid \gamma) = H(\alpha, \gamma) + H(\beta, \gamma) - H(\alpha, \beta, \gamma) - H(\gamma)$.

Лемма 2.5. *Все определения условной взаимной информации эквивалентны.*

Доказательство. (3) \iff (4).

$$(3) = H(\alpha | \gamma) + H(\beta | \gamma) - H(\alpha, \beta | \gamma) = H(\alpha, \gamma) - H(\gamma) + H(\beta, \gamma) - H(\gamma) - H(\alpha, \beta, \gamma) + H(\gamma).$$

□

Применение: опять о поиске фальшивой монетки

Теперь у нас достаточно знаний, чтобы доказать, что за три взвешивания нельзя найти одну фальшивую монету из 14, даже если не нужно определять её относительный вес.

Доказательство. Предположим, что существует способ найти фальшивую монету за три взвешивания. Тогда протокол взвешивания можно представить в виде полного тричного дерева, где каждый лист помечен номером монетки, которая оказалась фальшивой (у нас как раз ровно $3^3 = 27$ исходов).

Давайте введём следующее распределение вероятностей α . Пусть монета, номер которой находится в листе, соответствующем трём равенствам (такой лист только один), имеет номер i . В нашем распределении вероятностей монета с номером i будет фальшивой с вероятностью $1/27$. Оставшиеся монеты оказываются фальшивыми с вероятностями $2/27$, причём с вероятностью $1/27$ монета оказывается легче, чем настоящая, и с такой же вероятностью она оказывается тяжелее настоящей.

$$H(\alpha) = \log 27 = 3 \log 3.$$

Пусть случайные величины $\beta_1, \beta_2, \beta_3$ соответствуют результатам первого, второго и третьего взвешивания соответственно. Значение α однозначно определяется после трёх взвешиваний: $H(\alpha | \beta_1, \beta_2, \beta_3) = 0$, а следовательно

$$H(\alpha) \leq H(\beta_1, \beta_2, \beta_3) \leq H(\beta_1) + H(\beta_2) + H(\beta_3) \leq 3 \log 3.$$

Таким образом каждое взвешивание должно иметь энтропию ровно $\log 3$. Рассмотрим первое взвешивание. Пусть на чашах весов лежит по k монет. Вероятность каждого исхода взвешивания ($<$, $>$, $=$) относительно распределения α должна быть ровно $1/3$.

$$\Pr[<] = \frac{k}{27} + \frac{k}{27} = \frac{1}{3}.$$

Таким образом $2k = 9$, а значит нет такого целого k . □

Кодирование

Однозначно декодируемые коды

Определение 3.1. Будем называть *кодом* функцию $C : \{a_1, a_2, \dots, a_n\} \rightarrow \{0, 1\}^*$, сопоставляющую буквам некоторого алфавита *кодовые слова*. Если любое сообщение, которое получено применением кода C , декодируется однозначно (т.е. только единственным образом разрезается на образы C), то такой код называется *однозначно декодируемым*.

Определение 3.2. Код называется *префиксным* (*беспрефиксным*, *prefix-free*), если никакое кодовое слово не является префиксом другого кодового слова.

Теорема 3.1 (Неравенство Крафта-Макмилана). *Для любого однозначно декодируемого кода со множеством кодовых слов $\{c_1, c_2, \dots, c_n\}$ выполняется следующее неравенство:*

$$\sum_{i=1}^n 2^{-|c_i|} \leq 1.$$

Лемма 3.1. *Для префиксных кодов верно неравенство Крафта-Макмилана.*

Доказательство. Рассмотрим дерево префиксного кода и посчитаем суммарную меру поддеревьев, которые соответствуют кодовым словам. \square

Утверждение 3.1. *Для префиксных кодов верно и обратное: если есть набор целых чисел $\{\ell_1, \ell_2, \dots, \ell_n\}$, удовлетворяющие неравенству Крафта-Макмилана*

$$\sum_{i=1}^n 2^{-\ell_i} \leq 1,$$

то существует префиксный код с кодовыми словами $\{c_1, c_2, \dots, c_n\}$, где $|c_i| = \ell_i$.

Доказательство. Отсортируем ℓ_i по возрастанию и будем развешивать их в бесконечном двоичном дереве, выбирая каждый раз самый левый свободный узел соответствующей меры. Можно заметить, что мы всегда сможем найти такой узел. \square

Следствие 3.1. *Для любого однозначно декодируемого кода существует префиксный код с теми же длинами кодовых слов.*

Доказательства теоремы 3.1. Сопоставим кодовым словам $\{c_i\}$ мономы $\{p_i\}$ от переменных x и y таким образом, что каждый '0' в кодовом слове соответствует x , а каждая '1' — y :

$$c_i = 0110101 \implies p_i(x, y) = xyuxyxy.$$

Рассмотрим следующее выражение для некоторого L .

$$\left(\sum_{i=1}^n p_i(x, y) \right)^L = \sum_{\ell=L}^{\max |c_i| \cdot L} M_\ell(x, y),$$

где M_ℓ обозначает сумму всех получившихся одночленов степени ℓ . Заметим, что в каждом M_ℓ не более 2^ℓ одночленов: в противном случае код не был бы однозначно декодируемым — каждый одночлен (без учёта коммутативности и ассоциативности) мог получиться не более одного раза.

Теперь рассмотрим значение этого выражения при $x = y = \frac{1}{2}$.

$$\left(\sum_{i=1}^n p_i\left(\frac{1}{2}, \frac{1}{2}\right) \right)^L = \sum_{\ell=L}^{\max |c_i| \cdot L} M_\ell\left(\frac{1}{2}, \frac{1}{2}\right) \leq \sum_{\ell=L}^{\max |c_i| \cdot L} (2^{-\ell} \cdot 2^\ell) \leq L \cdot \max |c_i| = O(L). \quad (1)$$

Предположим теперь, что неравенство Крафта-Макмилана не выполняется, т.е.

$$q = \sum_{i=1}^n p_i(1/2, 1/2) = \sum_{i=1}^n 2^{-|c_i|} > 1.$$

Сравнивая это с (1) получаем противоречие: $q^L = O(L)$ (левая часть растёт экспоненциально, а правая — линейно). \square

Пусть для каждого символа алфавита задана вероятность p_i . Нас будут интересовать самые короткие в среднем коды, т.е. такие, что

$$\sum_{i=1}^n p_i \cdot |c_i| \rightarrow \min.$$

Теорема 3.2 (Шеннон). *Для любого однозначно декодируемого кода выполняется*

$$\sum_{i=1}^n p_i \cdot |c_i| \geq \sum_{i=1}^n p_i \cdot \log \frac{1}{p_i}.$$

Доказательство. Перенесём всё в правую часть и применим неравенство Йенсена:

$$\sum_{i=1}^n p_i \cdot \log \frac{2^{-|c_i|}}{p_i} \leq \log \sum_{i=1}^n \left(p_i \frac{2^{-|c_i|}}{p_i} \right) = \log \sum_{i=1}^n 2^{-|c_i|} \leq \log 1 = 0.$$

\square

Теорема 3.3 (Шеннон). *Для любого распределения вероятностей $\{p_1, p_2, \dots, p_n\}$ существует однозначно декодируемый/префиксный код $\{c_1, c_2, \dots, c_n\}$, такой что*

$$\sum_{i=1}^n p_i \cdot |c_i| \leq \sum_{i=1}^n p_i \cdot \log \frac{1}{p_i} + 1.$$

Замечание 3.1. От '+1' в правой части никак не избавиться: например, если у нас только два символа в алфавите, то $\sum p_i \cdot |c_i| = 1$, в то время как $\sum p_i \log \frac{1}{p_i}$ может быть сколько угодно близко к нулю.

Доказательство. Покажем, что найдутся $\{c_1, c_2, \dots, c_n\}$ такие, что $|c_i| = \lceil \log \frac{1}{p_i} \rceil$. Код существует, т.к. для длин c_i выполняется неравенство Крафта-Макмилана:

$$\sum_{i=1}^n 2^{-|c_i|} = \sum_{i=1}^n 2^{-\lceil \log \frac{1}{p_i} \rceil} \leq \sum_{i=1}^n 2^{-\log \frac{1}{p_i}} = \sum_{i=1}^n p_i = 1.$$

Теперь оценим среднюю длину кода:

$$\sum_{i=1}^n p_i \cdot |c_i| = \sum_{i=1}^n p_i \cdot \lceil \log \frac{1}{p_i} \rceil < \sum_{i=1}^n p_i \cdot (\log \frac{1}{p_i} + 1) = \left(\sum_{i=1}^n p_i \cdot \log \frac{1}{p_i} \right) + 1.$$

\square

Код Шеннона-Фано

Упорядочим вероятности символов по убыванию: $p_1 \geq p_2 \geq \dots \geq p_n$. Уложим на прямой без пропусков отрезки длиной p_1, p_2, \dots, p_n и обозначим i -ый отрезок через S_i , а их объединение — через S . Коды тех букв a_i , для которых отрезок S_i попал в левую половину S , будут начинаться с '0', а коды тех букв, для которых отрезок S_i попал в правую часть S — с '1'. Центральный отрезок может не попасть целиком в одну из половин S . Если центральный отрезок является первым или последним, то начнём его код, соответственно, с '0' или '1'. В противном случае отнесём его в произвольную половину S . Далее применяем эту стратегию отдельно для букв из левой половины S и отдельно для правой половины S . Повторяем так пока не получим уникальные коды для всех символов.

Определение 3.3. Будем называть кодирование, при котором для некоторой константы c и для всех i выполняется $|c_i| \leq -\log p_i + c$, *сбалансированным*.

Теорема 3.4 (Шеннон). *Средняя длина кода Шеннона-Фано близка к энтропии, но не обязательно оптимальна:*

$$\sum_{i=1}^n p_i \cdot |c_i| = H + O(1).$$

Код Хаффмана

Определение 3.4. Будем строить код Хаффмана по индукции. При $n = 2$ коды $c_1 = \langle 0 \rangle$, $c_2 = \langle 1 \rangle$. При $n > 2$ будем предполагать, что вероятности упорядочены по убыванию $p_1 \geq p_2 \geq \dots \geq p_n$. Заменяем символы a_{n-1} и a_n на символ a'_{n-1} с вероятностью $p'_{n-1} = p_{n-1} + p_n$. Построим код Хаффмана для $n - 1$ символа. Для символов a_{n-1} и a_n возьмём коды $c_{n-1} = c'_{n-1}0$ и $c_n = c'_{n-1}1$.

Лемма 3.2. *Средняя длина кодового слова для кода Хаффмана оптимальна, т.е. не превосходит средней длины любого другого префиксного кода (а значит и любого однозначно декодируемого).*

Следствие 3.2. *Для кода Хаффмана выполняется неравенство из теоремы Шеннона 3.3.*

Замечание 3.2. На энтропию случайной величины иногда удобно смотреть как на среднюю длину кода Хаффмана.

Блочное кодирование

Для того, чтобы нивелировать неустранимую '+1' в средней длине кода, мы будем кодировать не отдельные символы, а блоки символов. Пусть каждый блок состоит из k

символов. Пусть случайные величины $\alpha_1, \alpha_2, \dots, \alpha_k$ распределены как α и соответствуют буквам в блоке.

$$H(\alpha_1, \alpha_2, \dots, \alpha_k) = \sum_{i=1}^k H(\alpha_i) = k \cdot H(\alpha).$$

Тогда по теоремам Шеннона получается следующее ограничение на среднюю длину кода символа в блоке:

$$H(\alpha) \leq (\text{средняя длина кода буквы в блоке}) \leq H(\alpha) + \frac{1}{k}.$$

При кодировании блоков длины 100 мы получаем отклонение от энтропии не более, чем на 0.01. Однако мы не можем применить код Хаффмена, т.к. на вход алгоритму его построения нужно было бы передать n^{100} частот символов.

Арифметическое кодирование

Мы построим код со следующим ограничением на среднюю длину:

$$\sum_{i=1}^n p_i \cdot |c_i| \leq \sum_{i=1}^n p_i \cdot \log \frac{1}{p_i} + 2,$$

что хуже, чем в теореме Шеннона.

Определение 3.5. Будем называть полуинтервал *стандартным*, если он имеет вид $[0.v0_2, 0.v1_2)$, где v — это некоторая последовательность битов, а числа записаны в двоичной системе счисления. Будем сопоставлять каждому стандартному интервалу $[0.v0_2, 0.v1_2)$ код v .

Для первой буквы кода на отрезке $[0,1]$ мы отложим слева направо непересекающиеся интервалы длины p_i . Пусть первая буква блока — это a_{i_1} , тогда для второй буквы кода мы внутри интервала соответствующего p_{i_j} повторим эту операцию (отложим непересекающиеся интервалы), но длины интервалом будут уже масштабированы с коэффициентом p_i . Повторим эту операцию k раз. Получившемуся интервалу в качестве его кода сопоставим код наибольшего стандартного интервала, который полностью содержится внутри него.

Утверждение 3.2. В интервале $[a, b)$ всегда найдётся стандартный интервал длины 2^{-k} , где $\frac{b-a}{4} < 2^{-k} \leq \frac{b-a}{2}$, т.е. длина кода любого интервала при арифметическом кодировании не превосходит $\log \frac{4}{b-a} = \log \frac{1}{p_i} + 2$.

Замечание 3.3. В случае Марковской цепи можно строить код с соответствующими условными вероятностями.

Блочные коды с ошибками

Пусть $\alpha_1, \alpha_2, \dots, \alpha_n$ — независимые одинаково распределённые на $\{a_1, a_2, \dots, a_k\}$ случайные величины с вероятностями p_1, p_2, \dots, p_k . Рассмотрим блочное кодирование, заданное функциями E_n и D_n :

$$E_n : \{a_1, a_2, \dots, a_k\}^n \rightarrow \{0, 1\}^{L_n},$$

$$D_n : \{0, 1\}^{L_n} \rightarrow \{a_1, a_2, \dots, a_k\}^n,$$

Определение 3.6. Вероятность ошибки ε_n — это вероятность следующего события: $[(\alpha_1, \alpha_2, \dots, \alpha_n) = (a_{i_1}, a_{i_2}, \dots, a_{i_n}) \mid D_n(E_n(a_{i_1}, a_{i_2}, \dots, a_{i_n})) \neq (a_{i_1}, a_{i_2}, \dots, a_{i_n})]$.

Теорема 3.5 (Шеннон). При блочном кодировании допускающем ошибки выполняются следующие соотношения.

1. Если $h > H(\alpha) = \sum_{i=1}^k p_i \log \frac{1}{p_i}$, то существуют функции (E_n, D_n) для $L_n = \lceil h \cdot n \rceil$, такие что $\varepsilon_n \rightarrow 0$ при $n \rightarrow \infty$.
2. Если $h < H(\alpha) = \sum_{i=1}^k p_i \log \frac{1}{p_i}$, то для любых функций (E_n, D_n) для $L_n = \lceil h \cdot n \rceil$ вероятность ошибки $\varepsilon_n \rightarrow 1$ при $n \rightarrow \infty$.

Определение 3.7. Будем называть слово $w = \langle a_{i_1}, a_{i_2}, \dots, a_{i_n} \rangle$ δ -типичным, если каждая буква a_j встречается в нём t_j раз, причём

$$\begin{cases} t_j \leq (p_j + \delta) \cdot n, \\ t_j \geq (p_j - \delta) \cdot n. \end{cases}$$

Лемма 3.3. Для $\delta = n^{-0.49} = \frac{n^{0.01}}{\sqrt{n}}$ вероятность не δ -типичного не превосходит ε_n , для $\varepsilon_n \rightarrow 0$.

Доказательство. Применить неравенство Чебышева

$$P[|X - \mu| \geq \delta n] \leq \frac{\sigma^2}{(\delta n)^2} = \frac{np_i(1-p_i)}{\delta^2 n^2} = O(n^{-0.02}).$$

□

Лемма 3.4. Для $\delta = n^{-0.49}$ количество δ -типичных слов не превосходит $2^{h \cdot n}$ (при достаточно больших n).

Доказательство. Давайте для начала рассмотрим слова определённого типа, в которых буква i встречается n_i раз, $n_1 + n_2 + \dots + n_k = n$. Сначала оценим количество слов типа, в котором $n_i = n \cdot p_i$. Таких слов

$$\frac{n!}{n_1! n_2! \dots n_k!}.$$

По формуле Стирлинга $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \cdot (1 + o(1))$.

$$\begin{aligned} \log \frac{n!}{n_1! n_2! \cdots n_k!} &\approx \log \frac{\text{poly}(n) \left(\frac{n}{e}\right)^n}{\text{poly}(n) \left(\frac{n_1}{e}\right)^{n_1} \cdots \left(\frac{n_k}{e}\right)^{n_k}} = \\ &= \log \left(\frac{n}{n_1}\right)^{n_1} \cdots \left(\frac{n}{n_k}\right)^{n_k} + O(\log n) = \sum_{i=1}^k \underbrace{np_i}_{n_i} \cdot \log \frac{1}{p_i} + O(\log n) < h \cdot n. \end{aligned} \quad (2)$$

Мы оценили это только для конкретного типа слов. Давайте оценим для произвольного δ -типичного слова с $n_i = n \cdot (p_i + \Delta_i)$, где $|\Delta_i| \leq \delta$. Тогда (2) изменится следующим образом:

$$\cdots = \sum_{i=1}^k n(p_i + \Delta_i) \cdot \log \frac{1}{p_i + \Delta_i} + O(\log n) = n \cdot \sum_{i=1}^k p_i \cdot \log \frac{1}{p_i} + O(\log n) + n \cdot O(\delta) < h \cdot n.$$

(Действительно, энтропия — это непрерывная функция, а значит при небольшом отклонении она изменяется на $c \cdot \max_i \Delta_i$, где c зависит от производной функции энтропии.) Итого общее количество δ -типичных слов можно оценить как количество типов умноженное на количество δ -типичных слов одного типа:

$$\text{poly}(n) \cdot 2^{n \cdot H(\alpha) + n \cdot O(\delta) + O(\log n)} < 2^{h \cdot n}.$$

□

Доказательство теоремы 3.5.

1. Если мы будем кодировать только δ -типичные слова, то по лемме 3.4 нам будет достаточно длины кода L_n , а вероятность всех не типичных слов будет стремиться к нулю.
2. Обозначим за $\hat{\varepsilon}_n$ вероятность ошибки при декодировании δ -типичных слов. Мы хотим показать, что $\hat{\varepsilon}_n \rightarrow 1$. Давайте рассмотрим конкретное δ -типичное слово $w = \langle a_{i_1}, a_{i_2}, \dots, a_{i_n} \rangle$. Пусть p'_1, p'_2, \dots, p'_n — это частоты букв a_1, a_2, \dots, a_n . Оценим вероятность появления w :

$$\Pr[\langle a_{i_1}, a_{i_2}, \dots, a_{i_n} \rangle = w] = p_1^{p'_1 \cdot n} \cdots p_k^{p'_k \cdot n} = 2^{-(\sum_i p'_i \log \frac{1}{p_i}) \cdot n} \leq 2^{-(\sum_i p_i \log \frac{1}{p_i}) \cdot n + O(\delta_n \cdot n)}.$$

Всего мы может корректно закодировать не более 2^{L_n} δ -типичных слов, т.е. вероятность корректно декодировать δ -типичное слово

$$1 - \hat{\varepsilon}_n \leq 2^{L_n} \cdot 2^{-H(\alpha) \cdot n + O(\delta_n \cdot n)} \leq 2^{h \cdot n + 1} \cdot 2^{-H(\alpha) \cdot n + O(\delta_n \cdot n)} \rightarrow 0.$$

Таким образом $\hat{\varepsilon}_n \rightarrow 1$. Вместе с леммой 3.3 получаем, что $\varepsilon_n \rightarrow 1$.

□

Замечание 3.4. Используя предыдущую теорему можно, например, получить альтернативное доказательство неравенства $H(\alpha, \beta) \leq H(\alpha) + H(\beta)$. В левой части стоит асимптотическая средняя длина кода при блоковом кодировании (α, β) , а справа сумма средних длин кодов при блоковом кодировании α и β отдельно друг от друга. Т.к. мы можем рассмотреть кодирование (α, β) как конкатенацию кодов для α и β , то неравенство выполняется.

Свойства распределений

Энтропийные профили

Утверждение 4.1. Для любого $h \geq 0$ существует распределение α : $H(\alpha) = h$.

Доказательство. Возьмём некоторое целое n : $0 \leq h \leq \log n$. Искомое распределение — это линейная комбинация распределений с вероятностями $(1, 0, \dots, 0)$ и $(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$. \square

Каким может быть совместное распределение двух случайных величин α и β ? Рассмотрим как может быть устроен *энтропийный профиль* $(H(\alpha), H(\beta), H(\alpha, \beta))$.

Утверждение 4.2. Для любых чисел $h_1, h_2, h_{12} \geq 0$, которые удовлетворяют следующим соотношениям

$$\begin{cases} h_{12} \leq h_1 + h_2 & \iff t_0 = I(\alpha : \beta) \geq 0, \\ h_2 \leq h_{12} & \iff t_1 = H(\alpha | \beta) \geq 0, \\ h_1 \leq h_{12} & \iff t_2 = H(\beta | \alpha) \geq 0. \end{cases}$$

существует пара случайных величин (α, β) с энтропийным профилем (h_1, h_2, h_{12}) .

Доказательство. Пусть ξ_0, ξ_1, ξ_2 — независимые случайные величины с энтропиями t_0, t_1, t_2 соответственно. Тогда $\alpha = (\xi_0, \xi_1)$ и $\beta = (\xi_0, \xi_2)$ будут искомыми величинами.

$$\begin{cases} H(\xi_0) = t_0 = h_1 + h_2 - h_{12}, \\ H(\xi_1) = t_1 = h_{12} - h_2, \\ H(\xi_2) = t_2 = h_{12} - h_1. \end{cases} \quad \alpha \left(\begin{array}{ccc} & \xi_1 & \\ & \cap & \\ \xi_0 & & \xi_2 \\ & \cap & \\ & \beta & \end{array} \right)$$

\square

Давайте попробуем разобраться с аналогичным вопросом для троек случайных величин. Энтропийный профиль для тройки (α, β, γ) будет задаваться 7 числами:

$$(H(\alpha), H(\beta), H(\gamma), H(\alpha, \beta), H(\alpha, \gamma), H(\beta, \gamma), H(\alpha, \beta, \gamma)).$$

Для случайных величин (α, β, γ) можно записать 9 независимых неравенств.

$$\begin{aligned} H(\alpha | \beta, \gamma) \geq 0, & \quad I(\alpha : \beta) \geq 0, & \quad I(\alpha : \beta | \gamma) \geq 0, \\ H(\beta | \gamma, \alpha) \geq 0, & \quad I(\beta : \gamma) \geq 0, & \quad I(\beta : \gamma | \alpha) \geq 0, \\ H(\gamma | \alpha, \beta) \geq 0, & \quad I(\gamma : \alpha) \geq 0, & \quad I(\gamma : \alpha | \beta) \geq 0. \end{aligned}$$

Определение 4.1. Определим общую информацию трёх случайных величин

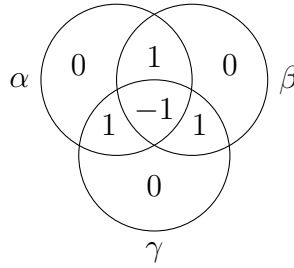
$$I(\alpha : \beta : \gamma) = I(\alpha : \beta) - I(\alpha : \beta | \gamma).$$

Утверждение 4.3. *Общая информация трёх случайных величин может быть отрицательной.*

Доказательство. Пусть α и β будут независимыми равномерно распределёнными на $\{0, 1\}$ случайными величинами. Случайная величина γ будет принимать значение из $\{0, 1\}$ в соответствии со следующим соотношением:

$$\alpha \oplus \beta \oplus \gamma = 0.$$

Мы получим следующую картину:

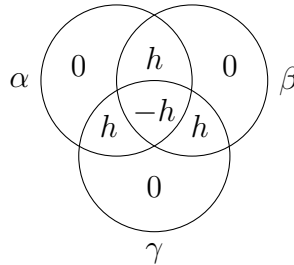


□

Утверждение 4.4. *Других неравенств для троек нет.*

Утверждение 4.5. *Есть профили, которые не реализуются никакими распределениями, но их мера 0.*

Упражнение 4.1. Доказать, что следующий профиль реализуется только при $h = \log n$ для некоторого целого n .



Утверждение 4.6. $2H(\alpha, \beta, \gamma) \leq H(\alpha, \beta) + H(\alpha, \gamma) + H(\beta, \gamma).$

Следствие 4.1 (Теорема 1.3). *Для $A \subset \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^*$*

$$2\chi(A) \leq \chi_{12}(A) + \chi_{13}(A) + \chi_{23}(A).$$

Доказательство. Пусть (α, β, γ) равномерно распределены на A .

$$2\chi(A) = 2H(\alpha, \beta, \gamma) \leq \underbrace{H(\alpha, \beta)}_{\leq \chi_{12}(A)} + \underbrace{H(\alpha, \gamma)}_{\leq \chi_{13}(A)} + \underbrace{H(\beta, \gamma)}_{\leq \chi_{23}(A)}.$$

□

Неравенства о тройках

Будем в различных предположениях доказывать следующее утверждение

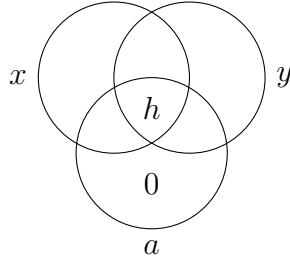
$$H(a | x) + H(a | y) \leq H(a).$$

Утверждение 4.7. Если a, x, y такие, что

$$\begin{cases} H(a | y, x) = 0, \\ I(x : y | a) = 0. \end{cases}$$

то $H(a | x) + H(a | y) \leq H(a)$.

Доказательство. Получается, что нам нужно доказать неотрицательность h .



Т.к. $I(x : y | a) = 0$, то $h = I(x : y) \geq 0$. □

Утверждение 4.8. Если a, x, y такие, что $H(a | y, x) = 0$ и

$$\begin{cases} A_i \sim X_j \\ A_i \sim Y_k \end{cases} \implies A_i \sim (X_i, Y_k),$$

то $H(a | x) + H(a | y) \leq H(a)$. (Обозначение $A_i \sim X_j \iff \Pr[a = A_i \wedge x = X_j] > 0$.)

Замечание 4.1. Условие $H(a | x, y) = 0$ можно интерпретировать так: $a = f(x, y)$.

Доказательство. Построим новое распределение (a', x', y') :

- a' имеет то же распределение, что и a ,
- условное распределение x' при условии a' совпадает с условным распределением x при условии a ,
- условное распределение y' при условии a' совпадает с условным распределением y при условии a ,
- x' и y' независимы при заданном a' .

$$\Pr[a' = A_i, x' = X_j, y' = Y_k] = \Pr[a' = A_i] \cdot \Pr[x' = X_j \mid a' = A_i] \cdot \Pr[y' = Y_k \mid a' = A_i].$$

Таким образом

$$H(a', x', y') = H(a') + H(x' \mid a') + H(y' \mid a') - \underbrace{I(x' : y' \mid a')}_0.$$

С другой стороны

$$H(a', x', y') \leq H(x') + H(y') + H(a' \mid x', y').$$

Кроме того, мы может стереть штрихи почти везде.

$$H(x) + H(y) + H(a' \mid x', y') \geq H(a', x', y') = H(a) + H(x \mid a) + H(y \mid a).$$

Покажем, что $H(a' \mid x', y') = 0$, т.е. $a' = f(x', y')$. Действительно: если тройка (A_i, X_i, Y_k) в новом распределении встречается с положительной вероятностью, то и в исходном распределении она так же встречалась с положительной вероятностью, следовательно $a' = f(x', y')$. Получаем: $H(a) + H(x \mid a) + H(y \mid a) \leq H(x) + H(y)$. Прибавим $H(a)$ к обеим частям неравенства:

$$H(x, a) + H(y, a) \leq H(x) + H(y) + H(a) \implies H(a \mid x) + H(a \mid y) \leq H(a).$$

□

Задача 4.1 (Верещагин). Рассмотрим двудольный граф с вершинами (L, R) с цветными рёбрами. Все рёбра инцидентные одной вершине разноцветные, степень в левой доле не меньше n , в правой — не меньше m . Пусть известно, что для пары вершин $(x \in L, y \in R)$ есть не более одного общего цвета. Докажите, что количество цветов хотя бы $n \cdot m$.

Заметим, что одноцветные рёбра образуют паросочетания. Для каждого цвета c соединим все согласованные с c вершины слева с согласованными с c вершинами справа. Получим биклику из рёбер цвета c .

Рассмотрим распределение на тройках (a, x, y) (цвет, вершина из левой доли, вершина из правой доли): выбираем цвет пропорционально размеру соответствующей биклики и выбираем случайное ребро этого цвета. Можно проверить, что выполняется следующее соотношение:

$$\begin{cases} A_i \sim X_j, \\ A_i \sim Y_k, \end{cases} \implies A_i \sim (X_i, Y_k).$$

Теперь применим: $\underbrace{H(a \mid x)}_{\geq \log n} + \underbrace{H(a \mid y)}_{\geq \log m} \leq H(a) \leq \log(\# \text{ цветов}).$

Криптография

Шифрования с закрытым ключом

Рассмотрим задачу кодирования сообщения при помощи симметричного шифрования. Будем считать, что вычислительные ресурсы противника неограниченны. Предположим, что мы шифруем сообщение x с ключом шифрования k . При шифровании сообщения мы получаем *шифrogramму* $m = E(x, k)$. Получатель шифrogramмы тоже знает ключ k и может узнать исходное сообщение $x = D(m, k)$.

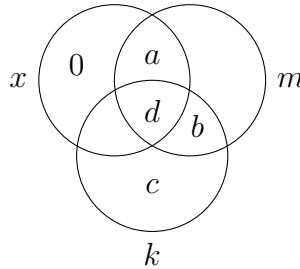
Будем предполагать, что x и k являются случайными величинами. Противник не знает x и k , но знает m . Для идеальной схемы шифрования должны выполняться следующие соотношения:

$$\begin{cases} H(m | x, k) = 0, \\ H(x | m, k) = 0, \\ I(m : x) = 0. \end{cases}$$

Теорема 5.1 (Шеннон). $H(k) \geq H(m)$, даже если мы ослабим условие выкинув первое условие $H(m | x, k) = 0$ (т.е. разрешим алгоритму E использовать случайные биты).

Замечание 5.1. Одноразовый блокнот (one-time notepad) обладает этим свойством.

Доказательство. По условию $a + d = 0$, т.е. $d = -a$.



Т.к. взаимная информация неотрицательна, то $d + b \geq 0$, т.е. $b \geq -d = a$. Теперь из $b \geq a$ и $c \geq 0$ следует $H(k) \geq H(x)$. \square

Схемы разделения секрета

Пусть у нас есть некоторый секрет S_0 и n участников и мы хотим разделить между ними этот секрет так, чтобы они могли им воспользоваться только все вместе, а любое подмножество участников — не могло.

Определение 5.1. *Совершенная схема разделения секрета* — это совместное распределение вероятностей $(S_0, S_1, S_2, \dots, S_n)$, такое что

$$\begin{cases} H(S_0 | S_1, S_2, \dots, S_n) = 0, \\ H(S_0 | S_{i_1}, S_{i_2}, \dots, S_{i_k}) = H(S_0), \quad k < n. \end{cases}$$

Второе условие можно переписать как $I(S_0 : S_{i_1}, S_{i_2}, \dots, S_{i_k}) = 0$.

Для совершенной схемы разделения секрета есть простая конструкция. Будем считать, что S_0 записан (закодирован) при помощи ℓ бит. Выберем независимо и равномерно $S_1, \dots, S_{n-1} \in \{0, 1\}^\ell$. S_n определяется из условия $S_0 \oplus S_1 \oplus S_2 \oplus \dots \oplus S_n = \vec{0}$ (покоординатная сумма по модулю 2).

Утверждение 5.1. *Предложенная схема разделения секрета является совершенной.*

Определение 5.2. *Пороговая совершенная схема разделения секрета* — это совместное распределение вероятностей $(S_0, S_1, S_2, \dots, S_n)$, такое что

$$\begin{cases} H(S_0 | S_{i_1}, S_{i_2}, \dots, S_{i_t}) = 0, \\ H(S_0 | S_{i_1}, S_{i_2}, \dots, S_{i_k}) = H(S_0), \quad k < t. \end{cases}$$

Пороговая схема Шамира. Будем считать, что секрет S_0 — это элемент некоторого конечного поля \mathbb{F}_q . Выберем случайный многочлен p над полем \mathbb{F}_q степени не более $t - 1$: выберем $t - 1$ коэффициент независимо и равномерно, а последний (свободный) коэффициент определим из соотношения $p(0) = S_0$. Выберем произвольным образом и сообщим всем участникам некоторый набор различных ненулевых элементов поля $a_1, a_2, \dots, a_n \in \mathbb{F}_q$ и вычислим секреты участников как значение полинома в соответствующих точках $S_i = p(a_i)$. Теперь любые t участников могут собраться, воспользоваться формулой для интерполяции многочлена и вычислить $S_0 = p(0)$. Если же соберётся меньше участников, то у них не будет никакой информации об S_0 .

Утверждение 5.2. *Пороговая схема Шамира является совершенной.*

Доказательство. Любой полином степени меньше $t - 1$ можно дополнить до полинома большей степени с любым значением в точке 0. \square

Определение 5.3. *Совершенная схема разделения секрета для структуры доступа* $\Gamma \subset 2^{[n]}$ (Γ должно быть замкнуто вверх) — это совместное распределение вероятностей $(S_0, S_1, S_2, \dots, S_n)$, такое что

$$\begin{cases} H(S_0 | S_{i_1}, S_{i_2}, \dots, S_{i_m}) = 0, & \{i_1, i_2, \dots, i_m\} \in \Gamma, \\ H(S_0 | S_{i_1}, S_{i_2}, \dots, S_{i_m}) = H(S_0), & \{i_1, i_2, \dots, i_m\} \notin \Gamma. \end{cases}$$

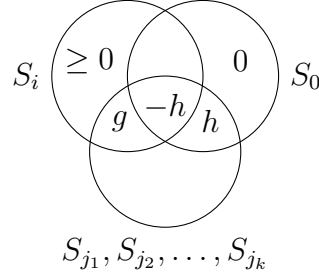
Определение 5.4. *Идеальная схема разделения секрета* — это совершенная схема разделения секрета с дополнительным требованием „экономности“.

$$\forall i \in \{1, 2, \dots, n\}, \quad H(S_i) \leq H(S_0).$$

Утверждение 5.3. *Если участник i является существенным в структуре доступа Γ (т.е. существует такое $s \in \Gamma$, что $s \setminus \{i\} \notin \Gamma$), то $H(S_i) \geq H(S_0)$.*

Замечание 5.2. Схема Шамира является идеальной.

Доказательство. Пусть $s = \{i, j_1, j_2, \dots, j_k\} \in \Gamma$, а $s \setminus \{i\} \notin \Gamma$. Обозначим взаимную информацию $I(S_0 : S_{j_1}, S_{j_2}, \dots, S_{j_k} | S_i)$ за h , а $I(S_i : S_{j_1}, S_{j_2}, \dots, S_{j_k} | S_0)$ за g . Из условия $I(S_0 : S_{j_1}, S_{j_2}, \dots, S_{j_k}) = 0$ получаем, что $I(S_0 : S_i : S_{j_1}, S_{j_2}, \dots, S_{j_k}) = -h$, аналогичным образом из $I(S_i : S_{j_1}, S_{j_2}, \dots, S_{j_k}) \geq 0$ получаем, что $g \geq h$.



Таким образом $H(S_i) \geq H(S_0)$. □

Замечание 5.3. Это утверждение показывает, что не бывает более „экономной“ схемы разделения секрета, чем идеальная.

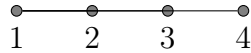
Утверждение 5.4. Для любой системы доступа Γ существует совершенная схема разделения секрета.

Доказательство. Давайте для каждого подмножества $A = \{i_1, i_2, \dots, i_k\} \in \Gamma$ создадим собственный набор секретов $S_{i_1}^A, S_{i_2}^A, \dots, S_{i_k}^A$: $S_{i_1}^A \oplus S_{i_2}^A \oplus \dots \oplus S_{i_k}^A = S_0$. (Достаточно рассматривать только минимальные множества A). □

Замечание 5.4. Предложенная схема не является идеальной.

Утверждение 5.5. Существуют структуры доступа, для которых не существует идеальной схемы разделения секрета.

Доказательство. Рассмотрим структуру доступа, заданную следующим графом (рёбра соответствуют авторизованным множествам).

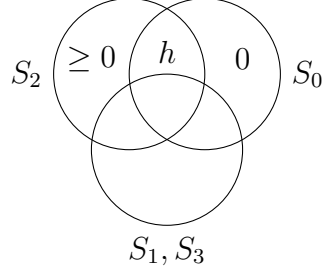


Покажем, что для этой структуры доступа $H(S_2) + H(S_3) \geq 3H(S_0)$, другими словами $\max_i \frac{H(S_i)}{H(S_0)} \geq 3/2$.

Для доказательства нам потребуются три леммы. Будем обозначать $h = H(S_0)$.

Лемма 5.1. $H(S_2 | S_1, S_3) \geq h$.

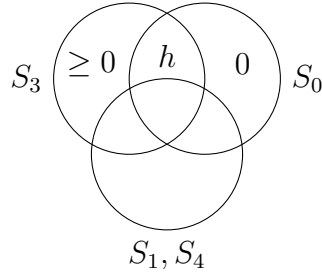
Доказательство. Второй участник может восстановить секрет, воспользовавшись либо секретом первого или секретом третьего участника, т.е. $I(S_2 : S_0 | S_1, S_3) = h$.



Таким образом $H(S_2 | S_0) \geq I(S_2 : S_0 | S_1, S_3) = h$. □

Лемма 5.2. $H(S_3 | S_1) \geq h$.

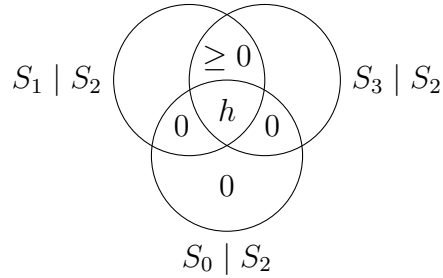
Доказательство. Аналогично предыдущей лемме получаем, что $H(S_3 | S_1, S_4) \geq h$, и как следствие $H(S_3 | S_1) \geq h$.



□

Лемма 5.3. $I(S_1 : S_3 | S_2) \geq h$.

Доказательство. Следующую схему следует интерпретировать как энтропия при условии S_2 .



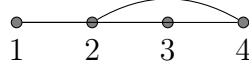
Заметим, что $I(S_1 : S_0 | S_2) = h$ и $I(S_3 : S_0 | S_2) = h$ в то время, как $I(S_1 : S_0 | S_2, S_3) = 0$ и $I(S_3 : S_0 | S_1, S_2) = 0$. Т.е. $I(S_1 : S_3 : S_0 | S_2) = h$, следовательно $I(S_1 : S_3 | S_2) \geq h$. □

Теперь осталось сложить результаты трёх лемм:

$$H(S_2) + H(S_3) \geq H(S_2, S_3) = H(S_2 | S_1, S_2) + H(S_3 | S_1) + I(S_1 : S_3 | S_2) + I(S_2 : S_1) \geq 3h.$$

□

Упражнение 5.1. Доказать, что для любой схемы разделения секреты для этой структуры $\max_i \frac{H(S_i)}{H(S_0)} \geq 3/2$.



Теорема 5.2 (Csirmaz'94). *Существуют структуры доступа Γ на n участниках, такие что для любой схемы разделения секрета $\max_i \frac{H(S_i)}{H(S_0)} \geq \Omega(n/\log n)$.*

Доказательство. Выберем n и k такие, что $n = 2^k + k + 1$, и два множества участников

$$A = \{a_1, a_2, \dots, a_k\},$$

$$B = \{b_1, b_2, \dots, b_{2^k-1}\}.$$

Для определения структуры доступа нам потребуются два семейства множеств. Пусть $\{A_0, A_1, A_2, \dots, A_{2^k-1}\}$ — это все подмножества A , причём $A_0 = A$ и для любых $i < j$ выполняется $A_i \not\subseteq A_j$ (например, можно их упорядочить по уменьшению размера). Построим множества $\{B_0, B_1, B_2, \dots, B_{2^k-1}\}$ следующим образом: $B_0 = \emptyset$, $B_i = \{b_1, b_2, \dots, b_i\}$. Теперь мы готовы определить структуру доступа Γ : $\Gamma = \{U_i\}_{i=0}^{2^k-1}$, где $U_i = A_i \cup B_i$.

Как и в предыдущих утверждениях обозначим $H(S_0)$ за h . В дальнейших рассуждениях мы будем использовать следующую нотацию: под энтропией некоторого множества участников $X = \{x_1, x_2, \dots, x_t\} \subset A \cup B$, мы будем понимать энтропию секретов, которые принадлежат участникам этого множества, т.е. $H(X) = H(S_{x_1}, S_{x_2}, \dots, S_{x_t})$.

Лемма 5.4. *Для $i = \{0, 1, 2, \dots, 2^k - 2\}$*

$$H(A \cup B_i) - H(B_i) \geq H(A \cup B_{i+1}) - H(B_{i+1}) + h.$$

Из этой леммы следует, что

$$\begin{aligned} H(A) &= H(A \cup B_0) - H(B_0) \geq H(A \cup B_1) - H(B_1) + h \geq \dots \geq \\ &\geq \underbrace{H(A \cup B_{2^k-1}) - H(B_{2^k-1})}_{\geq 0} + (2^k - 1) \cdot h. \end{aligned}$$

Получаем, что $H(A) = H(S_{a_1}, S_{a_2}, \dots, S_{a_k}) \geq (2^k - 1) \cdot h$. Следовательно есть i такое, что $H(S_{a_i}) \geq \frac{2^k-1}{k} \cdot h$. Вспомним, что мы выбрали $n = 2^k + k + 1$, т.е. $H(S_{a_i}) \geq \Omega(n/\log n) \cdot h$. Осталось доказать лемму.

Доказательство леммы 5.4. Докажем два неравенства:

1. $H(A_{i+1} \cup B_i) + H(B_{i+1}) \geq H(A_{i+1} \cup B_{i+1}) + H(B_i)$.
2. $H(A \cup B_i) + H(A_{i+1} \cup B_{i+1}) \geq H(A \cup B_{i+1}) + H(A_{i+1} \cup B_i) + h$.

□

Эта лемма завершает доказательство теоремы. □

Замечание 5.5. Нижние оценки на избыточную сложность совершенных схем разделения секрета влекут нижние оценки на схемную сложность монотонных функций.

Коммуникационная сложность

Пусть X , Y и Z — это три конечных множества, и пусть задана некоторая функция $f : X \times Y \rightarrow Z$. Два игрока, будем называть их Алиса и Боб, решают *коммуникационную задачу для функции f* , если:

1. множества X , Y , Z и функция f известны обоим игрокам,
2. Алиса знает некоторое $x \in X$,
3. Боб знает некоторое $y \in Y$,
4. Алиса и Боб стремятся вычислить $f(x, y)$.

Для решения этой коммуникационной задачи Алиса и Боб могут пересылать друг другу сообщения. Задача считается решённой, если оба игрока знают $f(x, y)$. Нас интересует минимальное количество битов, которое необходимо и достаточно переслать для вычисления $f(x, y)$.

Определение 6.1. *Коммуникационный протокол* для функции $f : X \times Y \rightarrow Z$ — это корневое двоичное дерево, которое описывает совместное вычисление Алисой и Бобом функции f . В этом дереве каждая внутренняя вершина v помечена меткой А или Б, означающей очередь хода Алисы или Боба соответственно. Для каждой вершины, помеченной А, определена функция $g_v : X \rightarrow \{0, 1\}$, которая говорит Алисе, какой бит нужно послать, если вычисление находится в этой вершине. Аналогично, для каждой вершины v с пометкой Б определена функция $h_v : Y \rightarrow \{0, 1\}$, которая определяет бит, который Боб должен отослать в этой вершине. Каждая внутренняя вершина имеет двух потомков, ребро к первому потомку помечено 0, а ребро ко второму потомку помечено 1. Каждый лист помечен значением из множества Z .

Вычисление по такому протоколу на конкретной паре входов (x, y) устроено так: изначально вычисление находится в корне. В каждой внутренней вершине v в зависимости от пометки либо Алиса, либо Боб пересылают один бит (он определяется соответствующей функцией g_v или h_v). После этого вычисление переходит в один из потомков вершины v по ребру, пометка которого совпадает с битом, переданным в вершине v . Когда вычисление приходит в лист, то оно завершается. Результат вычисления — это пометка в листе.

Будем говорить, что коммуникационный протокол *вычисляет функцию f* , если для всех пар $(x, y) \in X \times Y$ вычисление приходит в лист с пометкой $f(x, y)$. Теперь можно дать формальное определение *коммуникационной сложности функции f* .

Аналогичным образом можно определить *коммуникационный протокол, вычисляющий отношение $R \subset (X \times Y) \times Z$* — нужно только дополнительно потребовать, чтобы ответы Алисы и Боба были согласованы.

Определение 6.2. *Коммуникационная сложность* функции f определяется как наименьшая глубина протокола (максимальная рёберная длина пути от корня до листа), вычисляющего функцию f . Обозначается $D(f)$.

Утверждение 6.1. Для любой $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, $D(f) \leq n + 1$.

Доказательство. Алиса посылает Бобу свой вход, а Боб посылает Алисе значение f . \square

Пример 6.1. Примеры функций с нетривиальной верхней оценкой на коммуникационную сложность.

1. (Pointer Chasing) $D(PC) \leq k \log n$, где $PC(x, y) = \underbrace{x(y(x(y(x(y(x(y(x(0))))))))}_{k \text{ раундов}}$.

У игроков есть двудольный ориентированный граф на $2n$ вершинах, у которого исходящая степень каждой вершины равна 1. Алиса знает левую долю, Боб — правую. В начале они кладут фишку на вершину с номером 0 из доли Алисы и начинают передвигать её по рёбрам. Всего они должны сделать k переходов по рёбрам графа. Ответ — номер финальной вершины.

2. $D(MED) = O(\log^2 n)$, где x и y интерпретируются как характеристические функции подмножеств $[n]$, а $MED(x, y)$ — медиана их объединения.
3. $D(CIS_G) = O(\log^2 n)$, где x интерпретируется как характеристическая функция некоторой клики в графе G , а y — как характеристическая функция некоторого независимого множества в графе G . $CIS(x, y) = 1$, если клика и независимое множество имеют общую вершину. (Замечание: не известно графов G , для которых нельзя решить эту задачу за $O(\log n)$.)

Нижние оценки

Рассмотрим коммуникационный протокол для некоторой функции $f : X \times Y \rightarrow Z$. Для каждой вершины v определим множество $R_v \subset X \times Y$ — множество всех пар $(x, y) \in X \times Y$, для которых вычисление приходит в вершину v .

Утверждение 6.2. Для всех вершин v множество R_v является комбинаторным прямоугольником, т.е. существуют такие $X_v \subset X$ и $Y_v \subset Y$, что $R_v = X_v \times Y_v$.

Доказательство. Покажем по индукции. Это верно для корня. Если это верно для какой-то вершины v с пометкой A : $R_v = X_v \times Y_v$. Если Алиса пересылает бит b и вычисление переходит в вершину u , то $R_u = X_u \times Y_u$, где $X_u = \{x \in X_v \mid g_v(x) = b\}$, а $Y_u = Y_v$. Аналогично, если Боб посылает бит b и вычисление переходит в вершину u , то $R_u = X_u \times Y_u$, где $X_u = X_v$, а $Y_u = \{y \in Y_v \mid h_v(y) = b\}$. \square

Следствие 6.1. Листья коммуникационного протокола для функции f задают разбиение множества $X \times Y$ на одноцветные прямоугольники.

Будем обозначать $C^R(f)$ — минимальное количество одноцветных прямоугольников, покрывающих $X \times Y$.

Утверждение 6.3. $D(f) \geq \log C^R(f)$.

Доказательство. $D(f) \geq \log(\# \text{ листьев}) \geq \log C^R(f)$. \square

Метод размера прямоугольников. Определим некоторую весовую функцию на элементах $X \times Y$. Тогда верна следующая оценка

$$C^R(f) \geq \frac{w(X \times Y)}{\max_{\text{одноцв. } A \times B} w(A \times B)}.$$

Метод трудного множества (fooling set). Это частный случай метода размера прямоугольников, при котором фиксируется некоторое множество $F \subset X \times Y$, а $w(x, y)$ определяется следующим образом:

$$w(x, y) = \begin{cases} 1, & (x, y) \in F, \\ 0, & (x, y) \notin F. \end{cases}$$

При этом никакой прямоугольник не содержит более одного элемента из F . Следовательно $C^R(f) \geq |F|$.

Метод ранга матрицы. Рассмотрим матрицу функции f — матрицу, в которой строки индексированы элементами X , столбцы — элементами Y , а в ячейке (x, y) стоит $f(x, y)$. Если мы рассмотрим эту матрицу функции как матрицу M над некоторым довольно большим полем, то можно показать, что $C^R(f) \geq \text{rank } M$.

Упражнение 6.1. Докажите предыдущие утверждения.

Утверждение 6.4. $D(\text{EQ}) = n + 1$, где $\text{EQ}(x, y) = 1 \iff x = y$.

Утверждение 6.5. $D(\text{GE}) = n + 1$, где $\text{GE}(x, y) = 1 \iff x \geq y$.

Связь протоколов и формул

Определение 6.3. *Игра Карчмера-Вигдерсона для функции $f : \{0, 1\}^n \rightarrow \{0, 1\}$ — это следующая коммуникационная игра: Алиса получает $x \in f^{-1}(0)$, Боб получает $y \in f^{-1}(1)$, и они вместе пытаются найти такое $i \in [n]$, что $x_i \neq y_i$. Другими словами, игра Карчмера-Вигдерсона — это коммуникационная задача для отношения*

$$R_f = \{((x, y), i) \mid x \in f^{-1}(0), y \in f^{-1}(1), x_i \neq y_i\}.$$

Отношение R_f будем называть *отношением Карчмера-Вигдерсона* для функции f .

Определение 6.4. *Формула в базисе Де Моргана для функции $f : \{0, 1\}^n \rightarrow \{0, 1\}$ — это булева формула с переменными $\{x_1, x_2, \dots, x_n\}$, соответствующим отдельным битам входа f , и со связками $\{\wedge, \vee, \neg\}$, вычисляющая функцию f . Законы Де Моргана позволяют нам предполагать, что все \neg находятся непосредственно перед переменными. Заметим, что структура формулы Де Моргана представляет собой корневое дерево (листья соответствуют переменным, а внутренние вершина — логическим связкам).*

Будем называть *формульной сложностью* $L(f)$ функции $f : \{0, 1\}^n \rightarrow \{0, 1\}$ — это размер (количество вхождений переменных) минимальной формулы вычисляющей f . Если говорить более формально, то нужно говорить не о конкретной функции, а о последовательности функций.

Определение 6.5. Для функции $f : \{0, 1\}^* \rightarrow \{0, 1\}$ определим последовательность функций $\{f_1, f_2, \dots, f_n, \dots\}$, где $f_i : \{0, 1\}^i \rightarrow \{0, 1\}$ и $\forall x \in \{0, 1\}^i, f(x) = f_i(x)$. Тогда формульная сложность $L(f)$ функции f ограничена $g(n)$, если для любого n существует формула ϕ_n размера не более $g(n)$, вычисляющая функцию f_n .

Теорема 6.1 (Шеннон). *Существует $f : L(f) = \Omega(2^n/n)$.*

Доказательство. Посчитаем количество формул размера не более s (будем предполагать, что $s \geq n$): это количество можно оценить сверху как $(4 \cdot s)^s$. В то же время число всех функций $f : \{0, 1\}^n \rightarrow \{0, 1\}$ ровно 2^{2^n} . Какого s достаточно, чтобы вычислить все функции на n битах?

$$(4 \cdot s)^s \geq 2^{2^n} \implies s \cdot \log(4 \cdot s) \geq 2^n \implies s = \Omega(2^n/n).$$

□

Замечание 6.1. Этот подсчёт показывает, что существуют функции с экспоненциальной формульной сложностью. Более того, любая случайная функция с большой вероятностью имеет такую сложность.

Однако не известно *явных* функций большой сложности. Лучшая известная на данный момент нижняя оценка на формульную сложность явной функции это $\Omega(n^3)$ (оценка для функции Андреева, доказана Хостадом).

Теорема 6.2 (Карчмер-Вигдерсон). *Для каждой формулы ϕ вычисляющей f , существует такой протокол Π_ϕ для отношения Карчмера-Вигдерсона R_f , что его дерево совпадает с деревом, описывающим структуру формулу ϕ . Верно и обратное утверждение: если есть протокол для R_f , то есть и формула с такой же структурой.*

Доказательство. Ход Алисы будет соответствовать связке \wedge , ход Боба — связке \vee .

- **формула \rightarrow протокол**

Каждая внутренняя вершина протокола соответствует некоторой подформуле исходной формулы ϕ . Будем поддерживать следующий инвариант: пусть ϕ_v — подформула, соответствующая текущей вершине протокола v , тогда $\phi_v(x) = 0$, а $\phi_v(y) = 1$. Это верно для начальной вершины (т.к. верно для ϕ). Если для текущей вершины это верно, и $\phi_v = \phi_{v0} \wedge \phi_{v1}$, то Алиса пересылает бит b такой, что $\phi_{vb}(x) = 0$ (такой бит должен быть по свойствам \wedge , т.к. $\phi_v(x) = 0$). При этом мы знаем, что $\phi_v(y) = \phi_{v0}(y) = \phi_{v1}(y) = 1$, т.е. инвариант сохраняется. Аналогично, если $\phi_v = \phi_{v0} \vee \phi_{v1}$, то Боб пересылает бит b такой, что $\phi_{vb}(y) = 1$ (мы соответственно знаем, что $\phi_v(x) = \phi_{v0}(x) = \phi_{v1}(x) = 1$). Когда Алиса и Боб придут в некоторый лист, то по индукции получается, что значение в этом листе на входе Алисы отличается от значения в листе на входе Боба, а значит номер переменной в листе соответствует номеру бита различия.

• **протокол \rightarrow формула**

Будем последовательно строить формулы для внутренних вершин протокола от листьев к корню. При этом будем поддерживать следующий инвариант: пусть v — вершина протокола, $X_v \times Y_v$ — соответствующий прямоугольник, тогда формула ϕ_v для вершины v такая, что для всех $x \in X_v$, $\phi_v(x) = 0$ и для всех $y \in Y_v$, $\phi_v(y) = 1$. Пусть мы построили формулы ϕ_{v_0} и ϕ_{v_1} для сыновей некоторой вершины v . Если вершина v соответствовала ходу Алисы, то для всех входов Алисы из множества X_v формула ϕ_v должна быть равна 0. При этом по индукционному предположению мы знаем, что для некоторых входов Алисы (на которых Алиса посылает 0) $\phi_{v_0} = 0$, а для остальных обязательно $\phi_{v_1} = 0$. С другой стороны для всех входов Боба $y \in Y_v$, $\phi_{v_0}(y) = \phi_{v_1}(y) = 1$. Поэтому, если мы положим $\phi_v = \phi_{v_0} \wedge \phi_{v_1}$, то инвариант сохранится. Аналогично, если вершина соответствовала ходу Боба, то следует положить $\phi_v = \phi_{v_0} \vee \phi_{v_1}$.

Осталось объяснить, что мы будем делать с листьями. Заметим, что если в листе протокола написан некоторый индекс i , то в него могут попадать либо пары входов, для которых $(x_i = 0, y_i = 1)$, либо входы, для которых $(x_i = 1, y_i = 0)$, но не могут попадать одновременно. В противном случае можно было бы воспользоваться свойствами комбинаторных прямоугольников и дать Алисе и Бобу входы с одинаковыми i -ми битами, которые привели бы в этот же лист.

$$\begin{cases} (x, y) \in R_\ell, & x_i = 0, y_i = 1, \\ (x', y') \in R_\ell, & x'_i = 1, y'_i = 0. \end{cases} \implies (x', y) \in R_\ell.$$

Таким образом можно считать, что в каждом листе кроме номера бита различия записаны также значения этого бита у Алисы и у Боба. Если в листе ℓ с номером бита различия i записаны $(x_i = 0, y_i = 1)$, то $\phi_\ell = x_i$, в обратном случае $\phi_\ell = \neg x_i$.

□

Таким образом мы получили взаимно однозначное соответствие между протоколами и формулами. Проблема в том, что сложность протоколов мы до этого измеряли в терминах максимальной глубины, а сложность формул — в терминах количества листьев. Давайте определим сложность протокола в терминах количества листьев.

Определение 6.6. Для отношения R_f будем обозначать через $L(R_f)$ минимальное количество листьев в коммуникационном протоколе для R_f .

Следствие 6.2. Для любой функции f , $L(f) = L(R_f)$.

Определение 6.7. Пусть μ это некоторое распределение на входах Алисы и Боба, а X, Y — соответствующие случайные величины. *Внешнее информационное разглашение* протокола Π на распределении μ :

$$IC_\mu^{ext}(\Pi) = I(\Pi(X, Y) : X, Y).$$

Внутреннее информационное разглашение протокола Π на распределении μ :

$$IC_\mu^{int}(\Pi) = I(\Pi(X, Y) : X | Y) + I(\Pi(X, Y) : Y | X).$$

Лемма 6.1. Для любого протокола Π и любого распределения μ

$$D(\Pi) \geq \text{IC}_{\mu}^{\text{ext}}(\Pi) \geq \text{IC}_{\mu}^{\text{int}}.$$

Доказательство. Первое неравенство тривиально (нельзя раскрыть больше информации, чем количество переданных битов). Второе неравенство доказывается по индукции по количеству переданных бит. Пусть $T = \Pi(X, Y)$. База

$$I(\emptyset : X, Y) = I(\emptyset : X | Y) + I(\emptyset : Y | X).$$

Пусть бит номер i передаёт Алиса. Пусть T_i — это первые i битов, а b_i — бит пересылаемый Алисой. Тогда b_i полностью определяется по T_{i-1} и X .

$$\begin{aligned} I(T_i : X, Y) &= I(T_{i-1} : X, Y) + I(b_i : X, Y | T_{i-1}) \\ &= I(T_{i-1} : X, Y) + I(b_i : Y | T_{i-1}) + I(b_i : X | Y, T_{i-1}) \\ &\geq I(T_{i-1} : X, Y) + I(b_i : X | Y, T_{i-1}) \\ &\geq I(T_{i-1} : X | Y) + I(T_{i-1} : Y | X) + I(b_i : X | Y, T_{i-1}) \\ &= I(T_{i-1} : X | Y) + I(T_{i-1} : Y | X) + I(b_i : X | Y, T_{i-1}) + \underbrace{I(b_i : Y | X, T_{i-1})}_0 \\ &= I(T_i : X | Y) + I(T_i : Y | X). \end{aligned}$$

□

Теорема 6.3. Пусть Π коммуникационный протокол. Для любого распределения μ : $\log L(\Pi) \geq \text{IC}_{\mu}^{\text{ext}}(\Pi)$. Кроме того существует такое распределение μ^* для которого $\log L(\Pi) = \text{IC}_{\mu^*}^{\text{ext}}(\Pi)$. Будем называть μ^* труднейшим распределением для Π .

Доказательство. Для детерминированных протоколов $\text{IC}_{\mu}^{\text{ext}}(\Pi) = H_{\mu}(\Pi)$. Первое утверждение теоремы следует из верхней оценки на энтропию (энтропия случайной величины не превосходит логарифм числа исходов):

$$\text{IC}_{\mu}^{\text{ext}}(\Pi) = H_{\mu}(\Pi) \leq \log L(\Pi).$$

Для доказательства второго утверждения мы предъявим распределение μ^* : выберем (равномерно) случайный лист l протокола Π и в соответствующем прямоугольнике R_l выберем произвольную пару (x, y) . Полученное распределение μ^* равномерно на листьях Π , поэтому

$$\text{IC}_{\mu^*}^{\text{ext}}(\Pi) = H_{\mu^*}(\Pi) = \log L(\Pi).$$

□

Следствие 6.3. Пусть f — булева функция, $s \in \mathbb{N}$. $L(f) \geq s$ тогда и только тогда, когда для любого протокола Π для R_f существует распределение μ : $\text{IC}_{\mu}(\Pi) \geq \log s$.

Теорема 6.4 (Храпченко). $L(\oplus_n) \geq n^2$.

Доказательство. Покажем, что для любого протокола существует распределение μ : $IC_{\mu}^{ext}(\Pi) \geq 2 \log n$. Из этого напрямую следует, что $L(\oplus_n) \geq n^2$. Распределение μ будет равномерным распределением на парах вида $(x, x \oplus e_i)$, где $\oplus_n(x) = 0$, а строка e_i имеет единицу в позиции i и нули во всех остальных. Т.е., пары входов из распределения μ всегда будут отличаться только в одном бите.

$$IC_{\mu}^{ext}(\Pi) \geq IC_{\mu}^{int}(\Pi) = I(\Pi : X | Y) + I(\Pi : Y | X).$$

Рассмотрим одной из слагаемых $I(\Pi : X | Y)$.

$$\begin{aligned} I(\Pi : X | Y) &= H(X | Y) - H(X | Y, \Pi) \\ &= H(i | Y) - H(i | Y, \Pi) \\ &= \log n - 0. \end{aligned}$$

Таким образом $IC_{\mu}^{ext}(\Pi) \geq 2 \log n$. □

Упражнение 6.2. Докажите, что для любой булевой функции f и любого распределения μ существует протокол Π для R_f : $IC_{\mu}^{int}(\Pi) \leq 2 \log n$.

Упражнение 6.3. Будем называть *универсальным отношением* для строк длины n отношение $U_n = \{(x, y, i) \mid x, y \in \{0, 1\}^n, x_i \neq y_i\}$ (это обобщение понятия отношения Карчмера-Вигдерсона). Будем называть *расширенным универсальным отношением* для строк длины n отношение $U'_n = U_n \cup \{(x, x, \perp) \mid x \in \{0, 1\}^n\}$ (решая коммуникационную задачу для расширенного универсального отношения Алиса и Боб могут получить *одинаковые* строки и тогда они должны ответить \perp).

Докажите следующие утверждения:

1. $4 \cdot L(U_n) \geq L(U'_n) \geq L(U_n)$.
2. $L(U'_n) \geq 2^n$.

Упражнение 6.4. Пусть $f : \{0, 1\}^n \rightarrow \{0, 1\}$ некоторая булева функция. Определим функцию $(\vee_m \circ f) : \{0, 1\}^{m \times n} \rightarrow \{0, 1\}$ следующим образом:

$$(\vee_m \circ f)(x_1, x_2, \dots, x_m) = f(x_1) \vee f(x_2) \vee \dots \vee f(x_m),$$

где $x_i \in \{0, 1\}^n$ (т.е. мы определили композицию функция \vee_m и f). Докажите, что $L(\vee_m \circ f) = m \cdot L(f)$.

Алгоритмический подход

Колмогоровская сложность

Сколько информации в первых 10^{10} знаках числа π ? Её довольно мало, но сжать такое количество цифр, например, кодированием Хаффмена, не получится.

Определение 7.1. Частичная функция $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ называется *вычислимой*, если существует программа P :

- для $\forall x \in \text{dom } f: P(x)$ печатает $f(x)$,
- для $\forall x \notin \text{dom } f: P(x)$ не останавливается.

Определение 7.2. Пусть $F : \{0, 1\}^* \rightarrow \{0, 1\}^*$ — вычисляемая функция. *Сложность описания относительно F* определяется как

$$K_F(x) = \min\{|p| : F(p) = x\}.$$

Определение 7.3. Будем говорить, что способ описания F *не хуже* G , обозначается $F \prec G$, если существует константа c_G такая, что для $\forall x \in \{0, 1\}^*$

$$K_F(x) \leq K_G(x) + c_G.$$

Теорема 7.1 (Соломонова-Колмогорова). *Существует способ описания (вычисляемая функция) F такой, что для любого другого способа описания G выполняется $F \prec G$.*

Докажем сначала более простое утверждение.

Утверждение 7.1. Пусть F и G — два способа описания. Тогда существует способ описания H такой, что $H \prec F$ и $H \prec G$.

Доказательство. Определим H следующим образом: $H(0x) = F(x)$, $H(1x) = G(x)$ (если на каком-то входе x значение $F(x)$ или $G(x)$ не определено, то и H не определено на соответствующем входе $0x$ или $1x$). Тогда легко проверить, что для любых x верно $K_H(x) \leq K_F(x) + 1$ и $K_H(x) \leq K_G(x) + 1$. \square

Доказательство теоремы 7.1. Пронумеруем все программы натуральными числами (программ счётное число). Пусть F_N — это программа с номером N (для машин Тьюринга N называется *номером Гёделя*). Рассмотрим функцию $U(\langle N, x \rangle) = F_N(x)$, где пара $\langle N, x \rangle$ закодирована следующим образом $\underbrace{11\dots 1}_N 0x$. Тогда

$$K_U(x) \leq K_{F_N}(x) + N + 1.$$

(Для машин Тьюринга U — это универсальная машина Тьюринга.) \square

Определение 7.4. Будем называть $K(x) = K_U(x)$ *Колмогоровской сложностью x* .

Лемма 7.1. *Колмогоровская сложность обладает следующими свойствами.*

1. Существует c такая, что для всех x $K(x) \leq |x| + c$.
2. Существует c такая, что для всех x $K(xx) \leq |x| + c$.

3. Для любых оптимальных F_1 и F_2 выполняется $F_1 \prec F_2$ и $F_2 \prec F_1$, т.е. существует такая константа c , что $|K_{F_1} - K_{F_2}| \leq c$.

Доказательство. Третье свойство следует из определения. Докажем первые два.

1. Рассмотрим $H(x) = x$. Тогда $K(x) \leq K_H(x) + c = |x| + c$.

2. Рассмотрим $H(p) = pp$. Тогда $K(x) \leq K_H(xx) + c = |x| + c$.

□

Вопрос: может быть такая длина n , что для всех $x \in \{0, 1\}^n$ $K(x) < n$.

Утверждение 7.2. Для любого n существует $x \in \{0, 1\}^n$ такой, что $K(x) \geq n$ (т.е. x — несжимаемый).

Доказательство. Слов длины n всего 2^n . Слов сложности меньше n не больше, чем программ длины меньше n : $1 + 2 + \dots + 2^{n-1} = 2^n - 1 < 2^n$. □

Утверждение 7.3. Существует c такой, что для 99% слов длины n :

$$n - c \leq K(x) \leq n + c = |x| + c.$$

Доказательство. Второе неравенство мы уже доказали. Первое неравенство следует из того, что слов длины не более $n - c$ всего $1 + 2 + \dots + 2^{n-c} \leq 2^{n-c+1}$, т.е. доля таких слов не может быть больше 2^{-c+1} . При $c = 11$ доля таких слов меньше 0.01%. □

Утверждение 7.4. Не существует вычислимой функции $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$, которая была бы всюду определена и $f(\bar{n}) = x_n$, где $K(x_n) \geq n$ (\bar{n} означает двоичную запись числа n).

Доказательство. С одной стороны сложность x_n большая, с другой стороны мы можем описать x_n при помощи $\log n$ битов.

$$n \leq K(x_n) \leq K_f(\bar{n}) + O(1) \leq \log n + O(1).$$

□

Замечание 7.1. Это утверждение можно усилить, заменив „всюду определена“ на „определена для бесконечного числа входов“. Доказательство останется тем же.

Следствие 7.1. Отображение $x \rightarrow K(x)$ не является вычислимым.

Замечание 7.2. У этого факта есть довольно простое доказательство основанное на парадоксе Берри. Этот парадокс состоит в предложении рассмотреть

наименьшее натуральное число, которое нельзя определить фразой из не более чем четырнадцати русских слов.

Эта фраза содержит четырнадцать слов и определяет то самое наименьшее число, отсюда получаем противоречие. Аналогично, в предположении, что такое отображение является вычислимым, первую строку x для которой $K(x) \geq n$ мы можем описать при помощи $\log n$ битов.

Следствие 7.2. *Оптимальный способ описания не является всюду определённой функцией.*

Следствие 7.3. *Пусть есть некоторая формальная теория, т.ч. в ней можно записать ' $K(x) > c$ '. Для всех достаточно больших c и для всех x формулы ' $K(x) > c$ ' недоказуемы (и при этом почти все эти утверждения истинны).*

Доказательство. Если для любого c существует x такое, что ' $K(x) > c$ ' доказуемо, тогда перебирая все доказательства мы сможем по c построить x . \square

Следствие 7.4. *Первая теорема Гёделя о неполноте.*

Замечание 7.3. Это кроме всего прочего даёт способ с хорошей вероятностью порождать недоказуемые утверждения.

Утверждение 7.5. *Пусть $x = \langle 011010010 \dots 10110 \rangle$ длины n содержит $p \cdot n$ единиц и $(1 - p) \cdot n$ нулей, тогда*

$$K(x) \leq \left(p \cdot \log \frac{1}{p} + (1 - p) \cdot \log \frac{1}{1 - p} \right) \cdot n + O(\log n).$$

Доказательство. Рассмотрим следующее описание:

\langle количество '1', количество '0', номер перестановки с данным числом '1' и '0' \rangle .

Всего перестановок

$$C_n^{pn} = 2^{(p \cdot \log \frac{1}{p} + (1-p) \cdot \log \frac{1}{1-p}) \cdot n + O(\log n)}.$$

Т.е. $K(x) \leq \left(p \cdot \log \frac{1}{p} + (1 - p) \cdot \log \frac{1}{1-p} \right) \cdot n + O(\log n) = H(p) + O(\log n)$. \square

Замечание 7.4. В доказательстве важно кодировать эту тройку так, чтобы она однозначно разрезалась на три части. Можно, например, удвоить все биты первых компонент и добавить разделитель '01'.

Условная Колмогоровская сложность

Определение 7.5. Сложность *условного описания* x при условии y относительно F :

$$K_F(x | y) = \min\{|p| : F(p, y) = x\}.$$

Определение 7.6. Условное описание F не хуже, чем условное описание G , $F \prec G$, если существует c такая, что для любой x и y

$$K_F(x | y) \leq K_G(x | y) + c.$$

Теорема 7.2. Существует оптимальный способ описания условного описания F такой, что для любого другого способа условного описания G выполняется $F \prec G$.

Определение 7.7. Сложность оптимального описания x при условии y относительно оптимального способа условного описания $K(x | y)$ называется *условной Колмогоровской сложностью x при условии y* .

Утверждение 7.6. Условная Колмогоровская сложность обладает следующими свойствами.

1. $K(x | y) \leq K(x) + O(1)$.
2. $K(x | y) \leq |x| + O(1)$.
3. Существует такая константа c , что для всех n , всех y для 99% слов x длины n выполняется $|K(x | y) - n| \leq c$.
4. $K(x | x) = O(1)$.
5. Пусть f — вычислимая функция. Тогда существует c_f такая, что для всех x $K(f(x) | x) \leq c_f$.

Сложность пары

Будем обозначать сложность пары $K(x, y) = K(\langle x, y \rangle)$, где $\langle \cdot, \cdot \rangle$ — это произвольный вычислимый способ кодирования пар.

Утверждение 7.7. Следующее утверждение неверно:

$$\exists c \forall x, y K(x, y) \leq K(x) + K(y | x) + c.$$

Доказательство. Докажем от обратного. Пусть $|x| + |y| = n$. Тогда

$$K(x, y) \leq K(x) + K(y | x) + c \leq |x| + |y| + 2 \cdot O(1) + c = n + O(1).$$

С одной стороны различных пар всего $(n + 1) \cdot 2^n$. С другой стороны из оценки на сложность следует, что различных описаний пар не может быть больше $2^{n+O(1)}$. \square

Теорема 7.3. $\forall x, y K(x, y) \leq K(x) + K(y | x) + O(\log(|x| + |y|))$.

Доказательство. Рассмотрим следующий способ кодирования пар: $\langle \overline{|p|}01pq \rangle$, где $\overline{|p|}$ — это двоичная запись $|p|$, в которой удвоен каждый бит. \square

Теорема 7.4 (Колмогорова-Левина). $K(x, y) = K(x) + K(y | x) + O(\log(|x| + |y|))$.

Определение 7.8. *Взаимная информация x и y :*

$$I(x : y) = K(y) - K(y | x),$$

$$I(y : x) = K(x) - K(x | y).$$

Таким образом теорема Колмогорова-Левина — это теорема о симметрии взаимной информации.

$$I(x : y) = K(x) + K(y) - K(x, y) + O(\log(|x| + |y|)) = I(y : x).$$

Доказательство теоремы 7.4. Неравенство ‘ \leq ’ уже доказано. Осталось доказать ‘ \geq ’.

$$\underbrace{K(x)}_m - \underbrace{K(y | x)}_l \leq \underbrace{K(x, y)}_n + \underbrace{O(\log(|x| + |y|))}_{O(\log n)}.$$

Пусть $S = \{(a, b) \mid K(a, b) \leq n\}$. Заметим, что $(x, y) \in S$ и $|S| \leq 2^{n+1}$. Рассмотрим $S_x = \{(x, b) \mid (x, b) \in S\}$. По определению $(x, y) \in S_x$. Покажем, что

$$l = K(y | x) \leq \log |S_x| + O(\log n).$$

Будем перечислять множество S . В процессе этого перечисления мы будем получать точки из S_x . Для того, чтобы задать y нам нужно указать номер (x, y) в этом перечислении. Кроме того, чтобы такое перечисление запустилось, нам нужно знать число n . Получается, что

$$|S_x| \geq 2^{l - c \cdot \log n} \geq 2^{l'},$$

где l' — ближайшее снизу целое, т.е. $l' = \lfloor l - c \cdot \log n \rfloor$.

Посмотрим ещё раз на перечисление S . В процессе перечисления у нас возникают „тяжёлые сечения“ — те, в которых число элементов хотя бы $2^{l'}$. Для того, чтобы задать сечение S_x , нам нужно задать его порядковый номер в перечислении S среди всех „тяжёлых сечений“. Таким образом

$$m = K(x) \leq \log(\# \text{ тяжёлых сечений}) + O(\log n) + O(\log l').$$

Тяжёлых сечений не больше, чем $|S|/2^{l'}$.

$$m = K(x) \leq \log \frac{|S|}{2^{l'}} + O(\log n) = n - l + O(\log n).$$

Таким образом получаем утверждение теоремы: $m + l \leq n + O(\log n)$. □

Следствие 7.5. $|I(x : y) - I(y : x)| \leq O(\log K(x, y))$.

Замечание 7.5. Выберем n такое, что его двоичная запись несжимаема, т.е. $K(\bar{n}) = \log n + O(1)$. Возьмём $x \in \{0, 1\}^n$ такой, что $K(x | \bar{n}) = n + O(1)$. Тогда

- $I(\bar{n} | x) = K(x) - K(x | \bar{n}) = n + O(1) - (n + O(1)) = O(1)$,
- $I(x | \bar{n}) = K(\bar{n}) - K(\bar{n} | x) = (\log n + O(1)) - O(1) = \log n + O(1)$.

Т.е. нельзя уменьшить логарифмический зазор в теореме Колмогорова-Левина.

Метод несжимаемых объектов

Определение 7.9. *Конечный автомат с несколькими головками* — это конечный автомат, у которого на каждом шаге функция перехода по внутреннему состоянию автомата и по символам, на которых находятся головки, возвращает состояние на следующем шаге и номера головок, которые нужно сдвинуть, и при этом на каждом шаге сдвигается хотя бы одна головка.

Определим класс \mathcal{L}_k — класс языков, которые распознаются конечными автоматами с k головками.

Теорема 7.5. $\mathcal{L}_k \subsetneq \mathcal{L}_{k+1}$.

Определим следующее семейство языков над алфавитом $\{0, 1, \#\}$

$$A_n = \{w_1\#w_2\#\cdots\#w_n\#w_n\#\cdots\#w_1 \mid w_i \in \{0, 1\}^*\},$$

где $w_i \in \{0, 1\}^*$, $\forall i \in \{1, 2, \dots, n\}$.

При $n = 1$ для языка $A_1 = \{w_1\#w_1\}$ нужно две головки (по лемме о накачке конечный автомат с одной головкой этот язык не распознать).

При $n = 3$ можно распознать с четырьмя головками:

$$\begin{array}{cccc} w_1\#w_2\#w_3\#w_3\#w_2\#w_1. \\ \boxed{1} & & \boxed{2} & \boxed{3} & \boxed{4} \end{array}$$

Но можно обойтись и тремя головками (придумайте трюк):

$$\begin{array}{ccc} w_1\#w_2\#w_3\#w_3\#w_2\#w_1. \\ \boxed{1} & \boxed{2} & \boxed{3} \end{array}$$

Если использовать этот трюк для k головок, то можно было бы распознать язык A_n для $n \leq (k-1) + (k-2) + \cdots + 1$, т.е. $n \leq \frac{k \cdot (k-1)}{2}$. Таким образом конечный автомат с k головками распознаёт язык A_n для $n \leq \frac{k \cdot (k-1)}{2}$.

Лемма 7.2. A_n не распознаётся конечным автоматом с k головками, если $n > \frac{k \cdot (k-1)}{2}$.

Доказательство. Будем говорить, что пара головок (i, j) *инспектирует* w_ℓ , если найдётся шаг работы конечного автомата, когда i -ая головка читает символ левой копии w_ℓ , а j -ая головка читает символ правой копии w_ℓ .

Для любого $x \in A_n$ и для любой пары (i, j) существует не более одного блока w_ℓ такого, что пара (i, j) инспектирует w_ℓ . Если $n > k \cdot (k-1)/2$, то найдётся блок, который не инспектируется ни одной парой головок. Будем рассматривать некоторый $x \in A_n$ и предположим, что блок w_ℓ не инспектируется.

Замечание 7.6. Блок w_ℓ не инспектируется, поэтому, пока как какие-то головки находятся в левой копии w_ℓ , то в правой копии никакие головки находиться не могут.

Запишем *протокол работы автомата на слове x с выделенным ℓ* . Будем записывать состояние автомата каждый раз, когда происходят следующие события:

- вход головки в копию w_ℓ ,
- выход головки из копии w_ℓ .

Состояние автомата будет описываться внутренним состоянием автомата и позициями всех головок. Будем обозначать такой протокол $\pi(x, \ell)$.

Предположим, что для конкретного x

$$x = w_1 \# w_2 \# \dots \# w_\ell \# \dots \# w_n \# w_n \# \dots \# w_\ell \# \dots \# w_1,$$

конечный автомат не инспектирует блок ℓ . Рассмотрим вход x' с другим блоком w'_ℓ :

$$x' = w_1 \# w_2 \# \dots \# w'_\ell \# \dots \# w_n \# w_n \# \dots \# w'_\ell \# \dots \# w_1.$$

Утверждение 7.8. *Невозможно, что для x' блок ℓ тоже не инспектируется, и при этом протоколы равны $\pi(x, \ell) = \pi(x', \ell)$.*

Доказательство. Если протоколы равны, то автомат должен и допускать вход

$$x'' = w_1 \# w_2 \# \dots \# w_\ell \# \dots \# w_n \# w_n \# \dots \# w'_\ell \# \dots \# w_1.$$

Если какие-то головки находятся в w_ℓ , то автомат на x'' работает как на входе x . Если какие-то головки находятся в w'_ℓ , то автомат работает как на входе x' . Следовательно он должен принимать $x'' \notin A_n$. Таким образом мы пришли к противоречию. \square

Мы показали, что для разных x у нас должны быть разные протоколы. Таким образом зная ℓ и зная протокол мы можем восстановить w_ℓ — для этого нужно знать все остальные блоки и протокол. Наше наблюдение можно переписать следующим образом:

$$K(w_\ell \mid w_1, \dots, w_{\ell-1}, w_{\ell+1}, \dots, w_n, \ell, \pi(x, \ell)) = O(1).$$

Будем считать, что все блоки имеют длину N . Кроме того мы изначально потребуем, чтобы x был несжимаемым, т.е. $K(x) = K(w_1, w_2, \dots, w_n) \geq n \cdot N$. Тогда

$$n \cdot N \leq K(w_1, \dots, w_n) \leq \underbrace{(n-1) \cdot N}_{\{w_i\}_{i \neq \ell}} + \underbrace{O(\log n)}_{\ell} + \underbrace{4 \cdot k \cdot O(k \log nN)}_{\text{сложность } \pi(x, \ell)}.$$

При $N \rightarrow \infty$ мы получаем противоречие: $n \cdot N \leq (n-1)N + O(k^2 \log nN)$. \square

Доказательство теоремы 7.5. Язык $A_{\frac{k \cdot (k+1)}{2}}$ лежит в \mathcal{L}_{k+1} и не лежит в \mathcal{L}_k . \square