

Конспект по теории информации
IV семестр, 2021 год
Современное программирование, факультет математики и
компьютерных наук, СПбГУ
(лекции Дмитрия Соколова)

Тамарин Вячеслав

June 12, 2021

То есть фиксируем x_i, y_j , а z_k меняется в допустимых пределах. При этом $\sum_{i,j} K_{ij} = |A|$, так как это сумма размеров B_{ij} , которая равна размеру A .

Теперь рассмотрим множество

$$A_{13,23} = \{(x_i, z_s, y_j, z_l) \mid (x_i, z_s) \in A_{13}, (y_j, z_l) \in A_{23}\}.$$

Оценим размер множества $A_{13,23}$ сверху и снизу. Очевидно $|A_{13,23}| \leq |A_{13}| \cdot |A_{23}|$.

Теперь опишем элементы, которые заведомо есть в $A_{13,23}$, чтобы оценить его размер снизу. Для этого зафиксируем x_i, y_j , тогда все z_k из B_{ij} точно сюда подходят в качестве z_s или z_l . То есть таких четверок не меньше чем $|B_{ij}|^2 = K_{ij}^2$. А по всем i, j получается, что $|A_{13,23}| \geq \sum K_{ij}^2$.

Таким образом,

$$|A_{13}| \cdot |A_{23}| \geq |A_{13,23}| \geq \sum K_{ij}^2.$$

Домножим на $|A_{12}|$, которое не меньше количества K_{ij} (обозначим его за n), так как каждый набор B_{ij} соответствует одному элементу A_{12} :

$$\begin{aligned} |A_{12}| \cdot (|A_{13}| \cdot |A_{23}|) &\geq n \cdot \sum K_{ij}^2 \geq & (\text{неравенство о средних}) \\ &\geq \left(\sum K_{ij}\right)^2 = |A|^2 \end{aligned}$$

А это то, что мы и хотели доказать. ■

1.2 Угадывание числа

Обозначим $[n] := \{1, \dots, n\}$.

Симметричный вариант Пусть есть два игрока, первый загадывает число от 1 до n , а второй должен его угадать. Сколько вопросов необходимо задать, чтобы угадать число?

Есть два варианта игры:

- *Адаптивная*, когда второй игрок получает ответ на вопрос сразу
- *Неадаптивная*, когда он пишет сначала все вопросы, а потом получает на них все ответы.

Очевидно, что нам потребуется не менее логарифма запросов: нарисуем дерево, где вершины – запросы, по двум ребрам из которых можно перейти в зависимости от ответа. Листья должны содержать $[n]$, поэтому глубина дерева не менее логарифма.

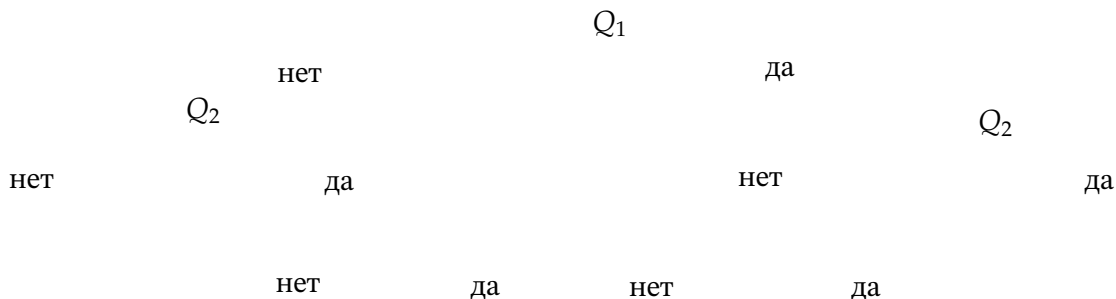


Figure 1.3: Граф вопросов

Теперь подумаем с точки зрения теории информации. Пусть h — число запросов, Q_i — ответ на i -ый запрос (один бит) по некоторому протоколу,

$$B := Q_1 \times \dots \times Q_h,$$

Мы хотим минимизировать h .

Рассмотрим $[n] \times B$ — все возможные пары по всем возможным значениям искомого числа и B . Нас интересует множество $A \subseteq [n] \times B$, соответствующее некоторым корректным запросам:

$$A = \{(m, b) \mid b = (q_1, \dots, q_h), m \text{ — согласовано с ответом}\}.$$

1. $\chi_{[n] \times B}(A) = 0$. Ответы на запросы должны однозначно определять число m . Это свойство говорит о корректности протокола, то есть нам ничего не нужно, чтобы, зная ответы, получить m .
2. $\chi(A) \geq \log n$, так как хотя бы столько мы запихнули.

$$\text{С другой стороны, } \chi(A) \leq \chi_B(A) + \chi_{[n] \times B}(A) \leq \chi(B) = \sum_{i=1}^h \chi(Q_i).$$

$$\text{А так как } \chi(Q_i) \leq 1, h \geq \log n.$$

Асимметричный вариант Пусть теперь за ответ «да» мы платим 1, а за «нет» 2. И мы хотим минимизировать не число запросов, а стоимость в худшем случае.

Будем делить запросом множество «пополам» с точки зрения стоимости.

Пусть Q_i — ответ на вопрос «верно ли, что загаданное число m лежит в множестве T_i ?

Пусть A_i — множество элементов, в котором может лежать m после первых $i - 1$ вопросов. В начале это все $[n]$, в конце — одно число.

$$A_i = \{a \in [n] \mid a \text{ согласовано с } Q_1, \dots, Q_{i-1}\}.$$

Стратегия минимальной цены бита информации: берем такое $T_i \subseteq A_i$, что

$$2(\chi(A_i) - \underbrace{\chi(T_i)}_{A_{i+1}}) = \chi(A_i) - \underbrace{\chi(A_i \setminus T_i)}_{A_{i+1}}.$$

Где разность $\chi(A_i) - \chi(A_{i+1})$ обозначает количество информации, которое мы получаем задав i -ый вопрос. То есть слева получили ответ «да» ($m \in T_i$), а справа «нет» ($m \in A_i \setminus T_i$).

Распишем по определению:

$$\begin{aligned} 2(\log|A_i| - \log|T_i|) &= \log|A_i| - \log|A_i \setminus T_i| \\ \log|A_i| &= 2\log|T_i| - \log|A_i \setminus T_i| \\ |A_i| &= \frac{|T_i|^2}{|A_i \setminus T_i|} \end{aligned}$$

Обозначим $|A_i| = k$, $|T_i| = t$ и решим уравнение:

$$\begin{aligned} k &= \frac{t^2}{k-t} \iff t^2 = k(k-t) = k^2 - kt \iff \\ t^2 + kt - k^2 &\iff t = \frac{-k \pm \sqrt{k^2 + 4k^2}}{2} = k \cdot \frac{-1 \pm \sqrt{5}}{2} = k \cdot \varphi \end{aligned}$$

То есть нужно выбирать T_i , чтобы $\varphi \cdot |T_i| = |A_i|$.

Тогда «средняя цена» бита будет равна $2(\chi(A_i) - \chi(T_i)) = 2 \log \frac{1}{\varphi}$

Докажем оптимальность. Пусть второй игрок меняет число, чтобы мы заплатили как можно больше, причем он знает нашу стратегию.

Если в нашем неравенстве знак \geq , он будет направлять на по «нет», а при \leq «да», за счет чего каждый бит он будет отдавать по цене большей, чем, если бы мы действовали в точности по стратегии.

Следовательно, любая другая стратегия будет требовать большего вклада.

Упражнение (Задача про взвешивания монеток). Есть n монеток и рычажные весы. Хотим найти фальшивую (она одна).

1. Пусть $n = 30$ и весы показывают, что больше, что меньше. Теперь запрос приносит $\log 3$ информации, так как три ответа.

$$\log 30 \leq \sum_{i=1}^h \chi(a_i) \leq h \log 3.$$

2. $n = 15$, но мы не знаем относительный вес фальшивой монеты. В прошлом неравенстве можно заменить 30 на 29. Если в какой-то момент у нас было неравенство, можем в конце узнать не только номер, но и относительный вес, поэтому у нас 29 исходов.
3. Вопрос: можно ли при $n = 14$? Нет.

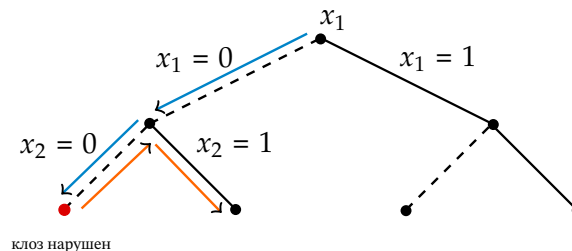
Лекция 2
8 April

1.3 Задача выполнимости

Вход: $\Phi = \bigwedge C_i$ — формула в КНФ.

Будем подставлять значения во все переменные по очереди, тем самым перемещаться по двоичному дереву.

Подставим $x_i = 0$. Если пока клозы не нарушены, подставляем далее $x_{i+1} = 0$ и проверяем клозы аналогично. Если же один из клозов нарушился, вернемся на шаг назад и подставим $x_i = 1$.



Это достаточно эффективный алгоритм, причем мы не ограничиваем выбор последовательности подстановок, порядок 0 и 1. Таким образом, если есть выполняющий набор, и мы на первом шаге подберем верное значение первой переменной, на второй для второй и т.д., то получим оптимальное решение. Однако не всегда такой алгоритм работает быстро. Пример: задача про рассадку голубей.

1.4 Рассадка голубей

У нас есть $n + 1$ голубь и n клеток, хотим показать, что нельзя рассадить в клетку по одному голубю.

Введем для каждой пары (голубь, клетка) переменную x_{ij} , которая равна 1, если i -ый голубь сидит в j -ой клетке, и $x_{ij} = 0$ иначе.

Тогда, чтобы рассадка была удачной, должны выполняться следующие условия:

- для всех $i \in [n + 1]$ верно $\prod_{j=1}^n (1 - x_{ij}) = 0$. То есть для каждого голубя нашлась клетка.
- для всех $i, i' \in [n + 1]$ и $j \in [n]$, где $i \neq i'$ верно $x_{ij} \cdot x_{i'j} = 0$. То есть никакие два голубя не сидят в одной клетке.

Пусть один игрок загадал расстановку голубей, а второй хочет найти дизъюнкт, для которого нарушается эта расстановка. Игра с монетками позволяет показать, что предыдущий алгоритм плохо работает на этой модели.

Chapter 2

Информация по Шеннону

2.1 Определения и свойства

На прошлой лекции поняли, что не всегда можем отличить некоторые множества.

Попробуем исправить данную ситуацию. Хотим понять состояния в Y , зная информацию об X . В среднем нам нужно сильно меньше информации, чем в крайнем случае.

Введем новую меру информации $\mu(\alpha)$, где α — распределение (множество и вероятности каждого элемента). Причем хотим, чтобы основные свойства были согласованы: ¹

1. $\mu(U_n) = \log n$;
2. $\mu(\alpha) \geq 0$;
3. $\mu(\alpha, \beta) = \mu(\alpha) + \mu(\beta)$, если α и β независимы.²

Если действовать как настоящие математики, можно переписать эти свойства в более общие:

1. *монотонность*: $\mu(U_M) \geq \mu(U_{M'})$, если $|M| \geq |M'|$;
2. *аддитивность*: $\mu(\alpha, \beta) = \mu(\alpha) + \mu(\beta)$, если α и β независимы;
3. *непрерывность*: $\mu(B_p)$ непрерывно по $p \in [0, 1]$, где B_p — распределение Бернулли для p .
4. *согласованность с условной вероятностью*:

$$\mu(B_p, \alpha) = \mu(B_p) + \Pr[B_p = 0] \cdot \mu(\alpha \mid B_p = 0) + \Pr[B_p = 1] \cdot \mu(\alpha \mid B_p = 1).$$

Этим аксиомам удовлетворяет примерно одна функция $\mu(X) := \sum p_i \log \frac{1}{p_i}$ с точностью до домножения на константу.

2.1.1 Энтропия

Определение 2: Энтропия

Для случайной величины α с вероятностями событий (p_1, p_2, \dots) меру

$$H(\alpha) = \sum_{i=1}^{|\text{supp}(\alpha)|} p_i \log \frac{1}{p_i}$$

будем называть **энтропией** и обозначать H .^a

¹ $\mu(x, y) = \mu((x, y))$

² (α, β) — распределение на парах.

Энтропия обозначает среднее по распределению α необходимое количество информации для записи элемента.

$\text{supp } \alpha$ — все возможные события, то есть имеющие ненулевую вероятность

Замечание. Энтропия равномерного распределения U_n равна $\log n$, так как вероятность каждого события $\frac{1}{n}$.

Замечание. Далее $H(p)$ обозначает энтропию для распределения нечестной монетки.

$$H(B_p) = H(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}.$$

Теорема 2.1.1. $H(\alpha) \leq \log |\text{supp}(\alpha)|$

□ Применим неравенство Йенсена

$$\sum_{i=1}^{|\text{supp}(\alpha)|} p_i \log \frac{1}{p_i} \leq \log \left(\sum_i p_i \frac{1}{p_i} \right) = \log |\text{supp}(\alpha)|$$

■

Теорема 2.1.2. $H(\alpha, \beta) \leq H(\alpha) + H(\beta)$

□

$$\begin{aligned} H(\alpha, \beta) &= \sum_{i,j} p_{i,j} \log \frac{1}{p_{i,j}} \\ H(\alpha) + H(\beta) &= \sum_i p_i \log \frac{1}{p_i} + \sum_j p_j \log \frac{1}{p_j} \end{aligned}$$

Заметим, что $p_i = \sum_j p_{i,j}$ и $p_j = \sum_i p_{i,j}$.

$$H(\alpha, \beta) - H(\alpha) - H(\beta) = \sum_{i,j} p_{i,j} \log \frac{1}{p_{i,j}} - \sum_i p_i \log \frac{1}{p_i} - \sum_j p_j \log \frac{1}{p_j} = \sum_{i,j} p_{i,j} \log \frac{p_i p_j}{p_{i,j}}.$$

Если α и β независимы, то все логарифмы обнуляются. Иначе по неравенству Йенсена

$$\sum_{i,j} p_{i,j} \log \frac{p_i p_j}{p_{i,j}} \leq \log \left(\sum_{i,j} p_i p_j \right) = 0.$$

■

2.1.2 Условная энтропия

Определение 3: Условная энтропия

Энтропией α при $\beta = b$ будем называть энтропию распределения α при условии, что $\beta = b$, то есть:

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

Тогда **энтропией α при условии β** назовем среднее значение по β энтропии α при $\beta = b$:

$$H(\alpha | \beta) := \mathbb{E}_{\beta \rightarrow b} H(\alpha | \beta = b) = \sum_b H(\alpha | \beta = b) \cdot \Pr[\beta = b].$$

Свойства.

1. $H(\alpha | \beta) \geq 0$

2. $H(\alpha | \beta) = 0 \iff \alpha$ однозначно определяется β



$$H(\alpha | \beta) = 0 \iff \forall b H(\alpha | \beta = b) = 0 \iff \forall b : \Pr[\alpha = i | \beta = b] = 1$$



3. $\forall f: H(\alpha | \beta) \geq H(f(\alpha) | \beta)$



Заметим, что достаточно показать неравенство $H(\alpha | \beta = b) \geq H(f(\alpha) | \beta = b)$, поэтому будем считать, что все дальнейшие рассуждения проводятся при условии $\beta = b$.

Пусть α принимает значения a_1, \dots, a_n с вероятностями p_1, \dots, p_n , а $f(\alpha)$ принимает значения f_1, \dots, f_k так, что для $\forall j f_i := f(a_{i_j})$ (разбили a_i -ые на k частей). Обозначим $P_i := \sum_j p_{i_j}$ - вероятность f_i .

$$\begin{aligned} H(\alpha | \beta = b) &= \sum_i \Pr[\alpha = i | \beta = b] \cdot \log \frac{1}{\Pr[\alpha = i | \beta = b]} && \text{(по определению } H) \\ &= \sum_i p_i \cdot \log \frac{1}{p_i} && \text{(по определению } p_i) \\ &= \sum_i \left(\sum_j p_{i_j} \cdot \log \frac{1}{p_{i_j}} \right) && \text{(перегруппировали слагаемые)} \\ &\geq \sum_i \left(\log \frac{1}{P_i} \cdot \sum_j p_{i_j} \right) && \text{(оценили максимум: } P_i \geq p_{i_j}) \\ &= \sum_i P_i \cdot \log \frac{1}{P_i} && \text{(свернули сумму в } P_i) \\ &= H(f(\alpha) | \beta = b) \end{aligned}$$



4. $H(\alpha, \beta) = H(\alpha) + H(\beta | \alpha) = H(\beta) + H(\alpha | \beta)$



$$\begin{aligned} H(\beta) + H(\alpha | \beta) &= \sum_j p_j \log \frac{1}{p_j} + \sum_j H(\alpha | \beta = j) \cdot \Pr[\beta = j] = \\ &= \sum_j p_j \log \frac{1}{p_j} + \sum_j \left(\sum_i \Pr[\alpha = i | \beta = j] \cdot \log \frac{1}{\Pr[\alpha = i | \beta = j]} \right) \cdot \Pr[\beta = j] = \\ &= \sum_j p_j \log \frac{1}{p_j} + \sum_j \left(\sum_i \Pr[\alpha = i, \beta = j] \cdot \log \frac{\Pr[\beta = j]}{\Pr[\alpha = i, \beta = j]} \right) = \\ &= \sum_j \sum_i p_{i,j} \log \frac{1}{p_j} + \sum_j \left(\sum_i p_{i,j} \cdot \log \frac{p_j}{p_{i,j}} \right) = \sum_{i,j} p_{i,j} \log \frac{1}{p_{i,j}} = H(\alpha, \beta) \end{aligned}$$

подсказка: $\Pr[A, B] = \Pr[A | B] \cdot \Pr[B]$



5. $H(\alpha, \beta) \geq H(\alpha)$



Очевидно из предыдущего свойства.



6. $H(\alpha) \geq H(\alpha | \beta)$



$$\begin{aligned}
 H(\alpha | \beta) - H(\alpha) &= \sum_j \sum_i \left(\Pr[\alpha = i | \beta = j] \log \frac{1}{\Pr[\alpha = i | \beta = j]} \right) \cdot \Pr[\beta = j] - \sum p_i \log \frac{1}{p_i} \\
 &= \sum p_{i,j} \log \frac{p_j}{p_{i,j}} - \sum p_{i,j} \log \frac{1}{p_i} \\
 &= \sum p_{i,j} \log \frac{p_i p_j}{p_{i,j}} \\
 &\leq \log \sum p_i p_j = \log 1 = 0
 \end{aligned}$$

(по нер-ву Йенсена)



7. Формула условной энтропии через отдельные вероятности



$$\begin{aligned}
 H(\alpha | \beta) &= \sum_j H(\alpha | \beta = b_j) \cdot \Pr[\beta = b_j] \\
 &= \sum_j \Pr[\beta = b_j] \cdot \sum_i \Pr[\alpha = a_i | \beta = b_j] \cdot \log \frac{1}{\Pr[\alpha = a_i | \beta = b_j]} \\
 &= \sum_{i,j} p_{i,j} \cdot \log \frac{p_j}{p_{i,j}}
 \end{aligned}$$



8. $H(\alpha | \beta) \geq H(\alpha | \beta, \gamma)$



$$\begin{aligned}
 H(\alpha | \beta) &= H(\alpha, \beta) - H(\beta) \\
 H(\alpha | \beta, \gamma) &= H(\alpha, \beta, \gamma) - H(\beta, \gamma) \\
 H(\alpha | \beta) - H(\alpha | \beta, \gamma) &= H(\alpha, \beta) - H(\beta) - (H(\alpha, \beta, \gamma) - H(\beta, \gamma)) \\
 &= H(\beta, \gamma) - H(\beta) - (H(\alpha, \beta, \gamma) - H(\alpha, \beta)) \\
 &= H(\gamma | \beta) - H(\gamma | \alpha, \beta)
 \end{aligned}$$



9. $2H(\alpha, \beta, \gamma) \leq H(\alpha, \beta) + H(\alpha, \gamma) + H(\beta, \gamma)$.



Заметим, что $H(\alpha, \beta, \gamma) = H(\alpha) + H(\beta, \gamma | \alpha) = H(\alpha) + H(\beta | \alpha) + H(\gamma | \alpha, \beta)$.

Теперь распишем аналогичное для правой части и воспользуемся $H(\alpha) \geq H(\alpha | \beta)$:

$$\begin{aligned}
 H(\alpha, \beta) &= H(\alpha) + H(\beta | \alpha) \\
 H(\alpha, \gamma) &= H(\alpha) + H(\gamma | \alpha) \geq H(\alpha) + H(\gamma | \alpha, \beta) \\
 H(\beta, \gamma) &= H(\beta) + H(\gamma | \beta) \geq H(\beta | \alpha) + H(\gamma | \alpha, \beta)
 \end{aligned}$$

Теперь если сложить все три формулы, то получим то, что надо.



2.2 Взаимная информация

Определение 4

Взаимная информация между случайными величинами α и β :

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

Взаимная информация в α и β при условии γ :

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

Свойства.

1. $I(\alpha : \beta) = I(\beta : \alpha).$

□ $I(\alpha : \beta) = H(\alpha) - H(\alpha | \beta) = H(\alpha) - H(\alpha, \beta) + H(\beta) = H(\beta) - H(\beta | \alpha) = I(\beta : \alpha)$

2. α и β независимы тогда и только тогда, когда $I(\alpha : \beta) = 0.$

□ $I(\alpha : \beta) = 0 \iff H(\alpha) = H(\alpha | \beta) \iff \text{независимы } \alpha \text{ и } \beta.$

3. $I(f(\alpha) : \beta) \leq I(\alpha : \beta)$ для любой функции $f.$

□ Для начала перепишем взаимную информацию:

$$\begin{aligned} I(\alpha : \beta) &= H(\alpha) - H(\alpha | \beta) \\ &= \sum_{i,j} p_{i,j} \log \frac{1}{p_i} - \sum_{i,j} p_{i,j} \log \frac{p_j}{p_{i,j}} \\ &= \sum_{i,j} p_{i,j} \log \frac{p_{i,j}}{p_i p_j} = \sum_{k,j} \sum_{i \in f^{-1}(k)} p_{i,j} \log \frac{p_{i,j}}{p_i p_j} \quad (\text{сгруппировали с помощью прообразов}) \end{aligned}$$

Аналогично

$$\begin{aligned} I(f(\alpha) : \beta) &= \sum_{k,j} \Pr[f(\alpha) = k, \beta = j] \log \frac{\Pr[f(\alpha) = k, \beta = j]}{\Pr[f(\alpha) = k] \Pr[\beta = j]} \\ &= \sum_{k,j} \left(\sum_{i \in f^{-1}(k)} p_{i,j} \right) \log \frac{\left(\sum_{i \in f^{-1}(k)} p_{i,j} \right)}{\left(\sum_{i \in f^{-1}(k)} p_i \right) p_j} \end{aligned}$$

Теперь для каждой пары k, j покажем неравенство отдельно

$$\begin{aligned} - \sum_{i \in f^{-1}(k)} p_{i,j} \log \frac{p_{i,j}}{p_i p_j} &= \sum_{i \in f^{-1}(k)} p_{i,j} \log \frac{p_i p_j}{p_{i,j}} \\ &= \left(\sum_{c \in f^{-1}(k)} p_{c,j} \right) \sum_{i \in f^{-1}(k)} \frac{p_{i,j}}{\left(\sum_{c \in f^{-1}(k)} p_{c,j} \right)} \log \frac{p_i p_j}{p_{i,j}} \\ &\quad (\text{поделили и домножили для того чтобы сумма коэффициентов была 1}) \\ &\leq \left(\sum_{c \in f^{-1}(k)} p_{c,j} \right) \log \left(\sum_{i \in f^{-1}(k)} \frac{p_i p_j}{\left(\sum_{c \in f^{-1}(k)} p_{c,j} \right)} \right) \end{aligned}$$

(по нер

≤

Chapter 3

Теория кодирования

Определение 5: Код

Будем называть **кодом** функцию $C: \Sigma = \{a_1, \dots, a_n\} \rightarrow \{0, 1\}^*$, сопоставляющую буквам некоторого алфавита кодовые слова.

Если любое сообщение, полученное применением кода C , однозначно декодируется, то такой код будем называть **однозначно декодируемым**.

3.1 Префиксные коды

Определение 6

Код называется **префиксным**, если никакое кодовое слово не является префиксом другого кодового слова.

Замечание. Очевидно, из этого следует однозначная декодируемость.

Теорема 3.1.1. Пусть набор целых чисел l_1, \dots, l_n удовлетворяет неравенству

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

Тогда существует префиксный код с кодовыми словами c_1, \dots, c_n , где $|c_i| \leq l_i$.

□ Индукция по n .

- База: $n = 1$. Очевидно.
- Переход: $n \rightarrow n + 1$. Пусть нам даны числа l_1, \dots, l_{n+1} .
 - Если среди них есть два одинаковых, $l_1 = l_2$, заменим их на одно $l = l_1 - 1$, чтобы сохранилась сумма обратных степеней двойки. Теперь можем применить предположение для n и найти префиксный код со словами c, c_3, \dots, c_{n+1} . Заменим слово c на $\bar{c}0$ и $\bar{c}1$, которые будут соответствовать l_1 и l_2 .
 - Если все различны, то можем выбрать слова длины l_1, l_2, \dots, l_n из слов вида $0, 10, 110, \dots$. Этот код будет префиксным, так как у всех разное число единиц перед нулем.

Теорема 3.1.2 (Неравенство Крафта-Макмиллана). Для любого однозначно декодируемого кода с кодовыми

словами c_1, \dots, c_n выполнено неравенство

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

□ Пусть $p_i(x, y)$ — моном соответствующий c_i^1 , и L — большое натуральное число.

Обозначим за $M_i(x, y)$ сумму всевозможных мономов степени i .

Рассмотрим

$$P^L(x, y) = \left(\sum_i p_i(x, y) \right)^L \leq \sum_{i=L}^{L \cdot \max(|c_i|)} M_i(x, y).$$

Неравенство выполняется, так как код однозначно декодируем, каждый моном в левой части есть и в правой.

Подставим $x = y = \frac{1}{2}$:

$$P_L\left(\frac{1}{2}, \frac{1}{2}\right) \leq \sum_{i=L}^{L \cdot \max(|c_i|)} (2^i \cdot 2^{-i}) \leq \mathcal{O}(L).$$

Если неравенство Крафта-Макмиллана не выполнено для данного кода, то

$$\sum_i p_i\left(\frac{1}{2}, \frac{1}{2}\right) = \sum_i 2^{-|c_i|} = 1 + \varepsilon > 1.$$

Тогда $P^L = (1 + \varepsilon)^L > \mathcal{O}(L)$, а это противоречит рассуждению о линейности роста. ■

Теорема 3.1.3. Любой однозначно декодируемый код можно переделать в префиксный с сохранением длин кодовых слов.

□ Пусть есть c_1, \dots, c_n — кодовые слова.

Для префиксного кода $\sum 2^{-|c_i|} \leq 1$, причем, если выполнено это неравенство, то есть префиксный код с такими длинами кодовых слов.

Докажем, что для любого декодируемого кода выполнено такое неравенство.

Построим многочлен для всех слов длины L . Для этого воспользуемся следующей идеей. Строка длины L переходит в конкатенацию кодовых слов для каждого символа исходной строки. Давайте в кодовых словах заменим 0 на x , а 1 на y . Тогда всем кодовым словам соответствуют некоторые мономы (например, $c_i = 010$ соответствует моном xux). Давайте будем считать, что x и y не коммутируют, чтобы удобнее было различать слова xux , соответствующие 010 и xxu , соответствующие 001. Будем говорить, что слову c_i соответствует $p_i(x, y)$. Тогда строка длины L есть произведение L таких мономов. Тогда коды всех слов длины L можно представить в виде многочлена $P(x, y)$, в котором операция сложения будет разделять разные коды

$$P(x, y) = \left(\sum_i p_i(x, y) \right)^L = \sum_{j=L}^{L \cdot \max |c_i|} M_j(x, y).$$

В $M_j(x, y)$ сгруппированны в сумму все мономы степени j , получающиеся при раскрытии скобок (то есть все различные слова длины j , получаемые из кодирования слов длины L). Так как код однозначно декодируемый, каждое слово является образом не более чем одной исходной строки. Тогда в $M_j(x, y)$ не более 2^j мономов.

Посчитаем $P\left(\frac{1}{2}, \frac{1}{2}\right)$.

¹Например, для $c_i = 010$, $p_i = xux$.

$$P\left(\frac{1}{2}, \frac{1}{2}\right) = \sum_{j=L}^{L \cdot \max |c_i|} M_j\left(\frac{1}{2}, \frac{1}{2}\right) \leq \sum_{j=L}^{\max_i c_i} 2^j \cdot 2^{-j} = \mathcal{O}(L).$$

Посчитаем еще раз по второму представлению

$$P\left(\frac{1}{2}, \frac{1}{2}\right) = \left(\sum_i 2^{-|c_i|} \right)^L.$$

Если сумма в скобках больше 1, получаем экспоненциальную оценку снизу. Следовательно, для больших N она обгонит линейную. Противоречие.

Как, используя это неравенство, построить префиксный код по декодируемому? Достаточно научиться по набору длин кодовых слов, для которого выполняется неравенство, построить префиксный код. Для этого давайте будем строить бинарное дерево. Изначально у нас есть только непомеченный корень и множество $A = \{l_1, \dots, l_n\}$ — длины кодовых слов. На каждом шаге мы смотрим наше множество A и проверяем, нет ли непомеченной вершины на глубине l_i . Если есть, помечаем ее и удаляем l_i из множества. После того, как таких вершин не осталось, раздваиваем все непомеченные вершины и повторяем алгоритм, пока множество не останется пустым. Нетрудно убедиться, что если выполнено неравенство, то для каждой длины найдется слово, так как если мы в вершине на высоте h запишем 2^{-h} , то сумма в листьях будет 1. А у нас кодовым словам соответствуют помеченные вершины, которые являются листьями (это, к слову, важно для беспрефиксности). ■

Теорема 3.1.4 (Шеннон). Для любого распределения p и однозначно декодируемого кода выполнено неравенство:

$$\sum_i p_i |c_i| \geq H(p).$$

□

$$\begin{aligned} H(p) - \sum_i p_i |c_i| &= \sum_i p_i \log \frac{2^{-|c_i|}}{p_i} \\ &\leq \log \sum_i p_i \cdot \frac{2^{-|c_i|}}{p_i} && \text{(Неравенство Йенсена)} \\ &\leq 0 && \text{(Неравенство Крафта-Макмиллана)} \end{aligned}$$

■

Теорема 3.1.5 (Шеннон). Для любого распределения p существует такой префиксный код, что^a

$$\sum_i p_i \cdot |c_i| \leq H(p) + 1.$$

^aЕдиница обязательно возникает, так как мы приводим непрерывную энтропию к дискретной величине

□ Угадаем длины кодов, чтобы выполнялось неравенство Крафта-Макмиллана. Пусть $|c_i| = \lceil \log \frac{1}{p_i} \rceil$, тогда неравенство из условия выполнено, так как $p_i |c_i| \leq p_i \log \frac{1}{p_i}$ и $\sum p_i = 1$. Кроме этого:

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

Теперь по [теореме 3.1.1](#) такой код действительно существует и удовлетворяет условию теоремы. ■

3.2 Примеры эффективных кодов

3.2.1 Код Шеннона-Фано

Отсортируем вероятности по убыванию $p_1 \geq p_2 \geq \dots \geq p_n$. Затем «уложим» их в отрезок $[0, 1]$, получая при этом такие точки:

$$0 \leq p_1 < p_1 + p_2 < \dots < p_1 + p_2 + \dots + p_n \leq 1.$$

Разделим отрезок $[0, 1]$ пополам и скажем, что слева кодовые слова начинаются с 0, справа с 1, а центральный p_i будет начинаться с нуля, если это p_1 , с единицы, если p_n , и, наконец, иначе выбираем любое значение.

Далее рекурсивно продолжаем процесс на группе нулей и на группе единиц.

Когда остался один кусок, останавливаемся.

Теорема 3.2.1.

$$\sum_0^n p_i \cdot |c_i| \leq H(p) + \mathcal{O}(1), \quad n \rightarrow \infty, \quad \mathcal{O}(1) \approx 3 \text{ или } 5.$$

Без доказательства. Дано как «упражнение со звездочкой».

3.2.2 Код Хаффмана

Опять отсортируем по убыванию $p_1 \geq p_2 \geq \dots \geq p_n$. Возьмем p_{n-1} и p_n . Заменяем их на один символ с вероятностью $p_n + p_{n-1}$, теперь продолжаем по индукции строить код для $n - 1$ символов.

Если объединенному символу соответствовал код \bar{c} , то для p_{n-1} задаем код $\bar{c}0$, а для p_n код $\bar{c}1$.

Теорема 3.2.2. Для кода Хаффмана выполнено неравенство:

$$\sum_{i=1}^n p_i |c_i| \leq H(p) + 1,$$

причем для любого другого однозначно декодируемого кода c'_i верно:

$$\sum_{i=1}^n p_i |c_i| \leq \sum_{i=1}^n p_i |c'_i|.$$

□ Достаточно доказать второе, а потом сравнить с кодом, который нам дает [теорема Шеннона 3.1.5](#), и получить нужное неравенство.

Рассмотрим набор c'_1, \dots, c'_n . Будем считать, что этот код префиксный, так как мы научились любой декодируемый код переделывать в префиксный с той же длиной кодовых слов.

- **Шаг 1.** Возьмем два минимальных c'_{n-1} и c'_n .

Заметим, что можно поменять их с символами максимальной длины c'_i и c'_j , при этом средняя длина кода не увеличится.

Значит, перестроили так, что код не ухудшился и c_n, c_{n-1} соответствуют символам с самой маленькой вероятностью и самой большой длиной кодовых слов.

- **Шаг 2.** Изучим коды c'_{n-1} и c'_n . Пусть они не имеют вид $\bar{v}0$ и $\bar{v}1$. И пусть $|c'_{n-1}| \leq |c'_n|$.

Посмотрим на c'_{n-1} : не умаляя общности он будет заканчиваться на 0 ($c'_{n-1} = \bar{s}0$).

Заменяем c'_n на $s1$. Что при этом могло сломаться?

Так как наш код префиксный, нам нужно проверить, что он префиксным и остался.

Заметим, что так как $c'_{n-1} = \bar{s}0$, префиксов s нет среди кодовых слов.

Единственная проблема тогда: среди кодовых слов может быть само $\overline{s1}$.

Тогда поступим следующим образом. Заменим c'_n на слово длины $|s| + 1$, а затем поменяем его местами с кодовым словом равным $\overline{s1}$.

Почему мы можем найти новое слово подходящей длины?

Воспользуемся неравенством Крафта-Макмиллана. Для целого q верно

$$\sum_i 2^{-|c_i|} = \frac{q}{2^{|c_{n-1}|}} + \frac{1}{2^{|c_n|}} \leq 1.$$

Но раз q целое, $\frac{q+1}{2^{|c_n|}} \leq 1$.

То есть мы уменьшили среднюю длину кода, сохранив при этом неравенство Крафта-Макмиллана.

Значит, перестроили так, что средняя длина кода не увеличилась и $c_{n-1} = \overline{v0}$, а $c_n = \overline{v1}$.

- **Шаг 3.** Заменим c_{n-1} и c_n на один символ с кодовым словом v . И применим предположение, что код Хаффмана оптимален для алфавита из $n - 1$ символа.

Тогда, раскрыв обратно, получим, что код Хаффмана оптимален и для n .



3.2.3 Арифметическое кодирование

Уложим вероятности аналогично на отрезок, при этом не обязательно в порядке убывания.

Назовем **стандартным** полуинтервал $[\overline{0.v0}, \overline{0.v1})$.

Найдем максимальный стандартный интервал в отрезке $[p_1 + \dots + p_{i-1}, p_1 + \dots + p_i]$. Пусть это $(0.v_i0, 0.v_i1)$.

Сопоставим i -ой букве код v_i0 .

Заметим, что такой код будет префиксным, так как отрезки не пересекаются, а, чтобы v_i было префиксом v_j , интервал $(0.v_j0, 0.v_j1)$ должен быть вложен в $(0.v_i0, 0.v_i1)$.

Лемма 2. В отрезке $[a, b]$ длина наибольшего стандартного интервала не меньше $\frac{b-a}{8}$.

□ Пусть $2^{-k-1} < \frac{b-a}{8} < 2^{-k}$. Рассмотрим все стандартные интервалы длины 2^{-k} . Заметим, что соседние интервалы находятся на расстоянии 2^{-k+1} .

Предположим, что ни один стандартный интервал не попал полностью в $[a, b]$. Тогда длина $[a, b]$ не больше суммы длин двух стандартных и расстояния между ними, то есть $2^{-k} \cdot 2 + 2^{-k+1} = 2^{-k+2}$.

Но $2^{-k+2} < b - a$. Противоречие. Следовательно, какой-то отрезок попал внутрь $[a, b]$. ■

Теорема 3.2.3. Для арифметического кода выполнено неравенство

$$\sum_i p_i |v_i| \leq H(p) + 2.$$

□ Из леммы 2 следует, что если $|v_i| = k$, то

$$0.v_i1 - 0.v_i0 = 2^{-k-1} \geq \frac{p_i}{8}.$$

Поэтому $k + 1 \leq \log \frac{8}{p_i}$, следовательно, $|v_i| = k \leq \log \frac{1}{p_i} + 2$ и

$$\sum_i p_i |v_i| \leq H(p) + 2(p_1 + \dots + p_n) \leq H(p) + 2.$$



3.3 Кодирование с ошибками

Пусть есть алфавит Σ размером k , «кодер» $E: [k]^n \rightarrow \{0, 1\}^{L_n}$ и «декодер» $D: \{0, 1\}^{L_n} \rightarrow [k]^n$.

Пусть есть распределение на буквах p_1, p_2, \dots, p_k .²

Откажемся от условия однозначного декодирования, но будем требовать, чтобы $\varepsilon_n := \Pr[D(E(x)) \neq x]$, где $|x| = n$ стремилось к нулю $\varepsilon_n \rightarrow 0$ при $n \rightarrow \infty$.³

Теорема 3.3.1. 1. Если $L_n > \lceil hn \rceil$, где $h > H(p)$, то кодирование есть, то есть существуют такие функции E и D , что $\varepsilon_n \rightarrow 0$.

2. Если $L_n < \lceil hn \rceil$, где $h < H(p)$, то $\varepsilon_n \rightarrow 1$ для любых E и D .

□ Зафиксируем $\delta = n^{-0.02}$.

Определение 7

Будем называть слово w **δ -типичным**, если

$$\forall i \left| \frac{n_i}{n} - p_i \right| \leq \delta, \quad n_i = \text{\#входа буквы } i.$$

• Докажем, что можем закодировать такие типичные слова.

- Пусть X_{ij} — характеристическая функция того, что в слове на позиции j находится буква i . Для случайной величины $X_i := \sum_j X_{ij}$ применим неравенство Чебышева:⁴

$$\Pr[|X_i - \mu| \geq \delta n] \leq \frac{\text{Var}[X_i]}{(\delta n)^2} = \frac{np_i(1-p_i)}{(\delta n)^2} = \mathcal{O}\left(\frac{1}{\delta^2 n}\right),$$

где $\mu = \mathbb{E}X_i = np_i$

Так как букв константное количество, вероятность нетипичности все равно останется очень маленьким и будет стремиться к нулю.

- Теперь докажем, что типичных слов не очень много. Количество слов, где буквы встречаются в количествах n_1, \dots, n_k равно

$$N_{n_1, \dots, n_k} = \frac{n!}{n_1! \cdot \dots \cdot n_k!}.$$

Обозначим $\delta_i = \frac{n_i}{n} - p_i$.

$$\begin{aligned} \log N &= \quad \quad \quad (\text{так как } n! = \text{poly}(n) \left(\frac{n}{e}\right)^n) \\ &= \log \left(\left(\frac{n}{n_1}\right)^{n_1} \cdot \left(\frac{n}{n_2}\right)^{n_2} \cdot \dots \cdot \left(\frac{n}{n_k}\right)^{n_k} \right) + \mathcal{O}(\log n) = \\ &= \sum_i n_i \log \frac{n}{n_i} + \mathcal{O}(\log n) = \quad \quad \quad (n_i \text{ по определению}) \\ &= n \sum_i (p_i + \delta_i) \cdot \log \frac{1}{p_i + \delta_i} + \mathcal{O}(\log n) \end{aligned}$$

- Теперь оценим число типичных слов.

Для этого можно посчитать для каждого распределения букв, сколько слов с таким распределением будут типичными.

²считаем, что слово состоит из независимых букв

³Если сделать равенство нулю, то особого сжатия не будет

⁴Здесь $\text{Var}[X_i]$ — дисперсия.

Число разбиений грубо оценивается сверху как n^k . А число слов в соответствующем разбиении можно оценить максимумом по всем δ_i , таким что $|\delta_i| \leq \delta$ (так как слово δ -типичное).

$$\begin{aligned}
 \log(\#(\delta\text{-типичных слов})) &\leq \log\left(n^k \cdot \max_{\delta_i} N\right) \leq \\
 &\leq \max_{\delta_i} H(p_1 + \delta_1, p_2 + \delta_2, \dots) \cdot n + \mathcal{O}(\log n) = \text{(Переход за кадром⁵)} \\
 &\leq n \cdot \max_i \sum_i (p_i + \delta_i) \cdot \log \frac{1}{p_i + \delta_i} + \mathcal{O}(\log n) \leq \\
 &\leq n \cdot \left(\max_i \sum_i p_i \log \frac{1}{p_i} + k\delta \max_i \log \frac{1}{p_i} \right) + \mathcal{O}(\log n) = \\
 &= nH(p) + \mathcal{O}(\delta \cdot n)
 \end{aligned}$$

Если теперь «кодер» может отобразить инъективно все типичные слова в набор битовых слов длины hn , при этом ошибаться он будет на нетипичных, количество которых стремиться к нулю.

- Во второй части докажем, что вероятность ошибки декодера на δ -типичных словах равна 1. Понятно, что этого будет достаточно.

Покажем, что вероятность того, что мы выкинем δ -типичное слово очень мала.

Пусть $L_n \leq hn$. Посмотрим на любое кодовое распределение слов. Покажем, что вероятность по нашему определению для δ -типичных слов больше, чем $\frac{1}{2^{L_n}}$.

$$\begin{aligned}
 \Pr[w] &= p_1^{n_1} \cdot p_2^{n_2} \cdot \dots \cdot p_k^{n_k} = \\
 &= 2^{-\sum_{i=1}^k (p_i + \delta) \log \frac{1}{p_i} n} \leq \\
 &\leq 2^{-H(p)n + \mathcal{O}(\delta n)}
 \end{aligned}$$

С какой вероятностью декодер декодер ответит правильно?

$$\begin{aligned}
 \Pr[D(E(w)) = w] &= \sum_x \Pr[D(x) = \\
 &= w \mid E(w) = x] \cdot \Pr[E(w) = x] \leq \sum_x \Pr[E(w) = x] \leq \\
 &\leq 2^{L_n} \cdot \max_w \Pr[w] \leq 2^{((L_n - H(p)) \cdot n + \mathcal{O}(\delta n))} \rightarrow 2^0
 \end{aligned}$$



Chapter 4

Коммуникационная сложность

Пусть у нас есть два игрока: Алиса и Боб. Они могут отправлять друг другу сообщения и хотят посчитать функцию (или отношение) $f: X \times Y \rightarrow Z$. Будем говорить, что они *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 $f(x, y)$.

4.1 Детерминированная модель

Определение 8: Коммуникационный протокол

Коммуникационный протокол для функции $f: X \times Y \rightarrow Z$ — корневое двоичное дерево, которое описывает совместное вычисление Алисой и Бобом функции f . В этом дереве:

- каждая внутренняя вершина v помечена меткой a или b , означающей очередь хода Алисы или Боба соответственно;
- для каждой вершины v , помеченной a , определена функция $g_v: X \rightarrow \{0, 1\}$, которая задает бит, который Алиса отправит Бобу, если вычисление будет находиться в v ; аналогично для каждой вершины v с пометкой b определена функция $h_v: Y \rightarrow \{0, 1\}$ для сообщений Боба;
- каждая внутренняя вершина имеет двух потомков, ребро к первому помечено нулем, ко второму — единицей;
- каждый лист помечен значением из множества Z .

Каждая пара входов (x, y) определяют путь от корня до листа в этом дереве. Будем говорить, что коммуникационный протокол **вычисляет** функцию f , если для всех пар $(x, y) \in X \times Y$ этот путь заканчивается в листе с пометкой $f(x, y)$.

Коммуникационной сложностью функции f называется наименьшая глубина протокола, вычисляющего f . Обозначается $D(f)$ или $D^{cc}(f)$.

Каждой функции можем сопоставить матрицу $X \times Y$, в каждой клетке (x_i, y_j) которой стоит значение $f(x_i, y_j)$.

4.1.1 Нижние оценки для детерминированного случая

Пусть наша функция $f: X \times Y \rightarrow Z$. Запишем для нее коммуникационную матрицу M размера $|X| \times |Y|$, где $M_{x,y} = f(x, y)$.

Рассмотрим такое подмножество R_v множества $X \times Y$, что $(x, y) \in R_v \iff$ протокол приводит в v .

Лемма 3. $R_v = X_v \times Y_v$ — комбинаторный прямоугольник.

Рассмотрим два доказательства:

□ Пусть (x, y) и (x', y') принадлежат R_v . Тогда (x, y') и (x', y) тоже принадлежат R_v , так как $a(x) = a(x')$ и $b(y) = b(y')$.

А из этого следует, что это комбинаторный прямоугольник. ■

□ Посмотрим на множество $X \times Y$ как на таблицу.

Пусть Алиса перешла по какому-то ребру. Вся таблица разделилась на две части по горизонтали: какие-то строки соответствуют $x \in X$, для которых Алиса отправляет 0, а какие-то для 1.

Если потом ход делает Боб, то он делит все текущие прямоугольники тоже на две части, но по вертикали.

И так далее.

В прямоугольнике для листа y всех элементов одинаковый ответ. То есть исходную матрицу можно разбить на комбинаторные прямоугольники, причем они естественно не пересекаются. ■

Рассмотрим величины $\chi_0(f)$ и $\chi_1(f)$, первая равна минимальному числу непересекающихся прямоугольников, которыми можно покрыть все нули в таблице, в вторая — все единицы.

Тогда листьев в двоичном дереве протокола будет хотя бы $\chi_0(f) + \chi_1(f)$. Следовательно, $D(f) \geq \log(\chi_0(f) + \chi_1(f))$.

Но эта оценка не всегда точна. Можно рассмотреть следующее разбиение на картинке 4.1.

Здесь $\chi_0(f) + \chi_1(f) = 4 + 1 = 5$. Заметим, что для такого разбиения не существует дерева протокола.

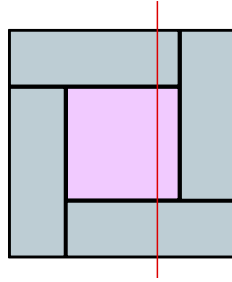


Figure 4.1: Пример неточной оценки

Посмотрим на первое действие игроков. Прямоугольник должен разделиться на две части, но любой разрез вдоль сторон разрежет один из внутренних прямоугольников.

Поэтому любой протокол не будет соответствовать этому разбиению. А тогда листьев будет больше пяти.

Теорема 4.1.1 (G,PW 16). Пусть χ — минимальное число одноцветных прямоугольников в разбиении. Существует f для которой

$$D(f) \geq \log^{2-\varepsilon} \chi(M_f).$$

Без доказательства.

4.2 Вероятностная модель

Теперь Алиса и Боб могут подбрасывать монетки. Либо эти монетки (оракулы) *публичны* (оба видят значения), либо *приватны* (тогда никто не видит, кроме пользователя).

Так как Алиса или Боб в случае публичного оракула, могут закрыть глаза на сообщения другого, публичный протокол не меньше приватного.

Скажем, что **протокол отработал корректно**, если

$$\forall x, y: \Pr_r[\pi(x, y) = f(x, y)] \geq \frac{2}{3}, \quad \pi(x, y) \text{ — результат работы.}$$

Определение 9

Будем говорить, что вероятностный протокол ε -**вычисляет** f , если для любой пары x, y с вероятностью r или s не менее $1 - \varepsilon$ результат протокола равен $f(x, y)$ (с точки зрения обоих игроков).

Через $R_\varepsilon(f)$ обозначается минимальная высота вероятностного протокола ε -вычисляющего f .

Через R_ε^{pub} — минимальная высота в случае, если $r = s$.

Лемма 4. $R_\varepsilon^{pub}(\text{EQ}) = \mathcal{O}(\log \varepsilon^{-1})$

4.3 Методы оценки коммуникационной сложности

4.3.1 Метод ранга

Пусть M_f — таблица некоторой функции f со значениями $\{0, 1\}$.

Обозначим за x_1, \dots, x_k листья дерева коммуникационного протокола для функции f , а R_{x_1}, \dots, R_{x_n} соответствующие им прямоугольники в M_f .

$$\text{rk}(M_f) \leq \sum_{i=1}^n \text{rk } R_{x_i} = \sum_{i=1}^n 1 = \# \text{ одноцветных прямоугольников в разбиении.}$$

Тогда коммуникационная сложность не меньше $\log \text{rk}(M_f)$.

Этот метод называется **методом ранга**.

Оценка для EQ Для функции $\text{EQ} =: \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, будет диагональная матрица. Поэтому одноцветных прямоугольников будет не меньше 2^n , а тогда коммуникационная сложность хотя бы n .

4.3.2 Fooling Set

Рассмотрим коммуникационную матрицу. Пусть мы хотим выбрать некоторое множество

$$S = \{(x_1, y_1), (x_2, y_2), \dots\},$$

такое что каждая пара точек не лежит в одном прямоугольнике.

Если две клетки в одном прямоугольнике, оставшиеся вершины тоже лежат в нем.

Тогда нужно для всех $i, j, i \neq j$ либо (x_i, y_j) , либо (x_j, y_i) покрашена в другой цвет.

Для EQ легко получить ту же оценку. Плюс, как как нужен хотя бы один лист для нуля, n не хватит, следовательно, $D(\text{EQ}) = n + 1$.

Теорема 4.3.1. Если существует Fooling set размера s , то $\text{rk}_R s \geq s$.

4.4 Связь со схемами. Теорема Шеннона

Определение 10

Формульная сложность $L(f)$ формулы f — минимальное возможное число листьев двоичного дерева, вычисляющего f .

Теорема 4.4.1 (Шеннон). Существует $f: \{0, 1\}^n \rightarrow \{0, 1\}$, такая что

$$L(f) \geq \Omega\left(\frac{2^n}{n}\right).$$

□ Всего функций такого вида 2^{2^n} , так как можно задать таблицей истинности.

Посчитаем число схем. Это ациклический граф и то, что записано в его узлах.

Пусть каждая вершина (S штук) выбирает себе двух предков. Так же в каждую вершину нужно что-то записать и на ребре можно ставить отрицание: хватит 3 бита. Еще есть входные данные (n штук).

Итого: $2^{S \cdot 2(\log S + \log n) + 3S}$.

Схем должно быть не меньше количества функций

$$2^{S \cdot 2(\log S + \log n) + 3S} \geq 2^{2^n}.$$

Отсюда получаем нужное неравенство. ■

Открытый вопрос: Можно ли предъявить $f \in \text{NP}$, что $L(f) \geq 10n$

4.5 Теорема Карчмера-Вигдерсона

Пусть нам дана $f: \{0, 1\}^n \rightarrow \{0, 1\}$.

Алиса получает число $x \in f^{-1}(1)$, а Боб $y \in f^{-1}(0)$. Их цель найти любой бит, в котором x и y отличаются.

Теорема 4.5.1 (Karchmer-Wigderson, 1990). $L(f)$ — размер минимальной формулы для f , тогда $L(f)$ — размер минимального протокола для KW_f

□

1 \implies 2 Нарисуем дерево вверх корнем. Также спустим все отрицания к листьям. Пусть в узле считается функция $f = g \vee h$, где g и h — соседи f .

Тогда $f(x) = 1$, $f(y) = 0$. Тогда $g(y) = h(y) = 1$, а для Алисы хотя бы одно из да

?

Покажем, что полученная сумма большая.

Зафиксируем i и распишем по определению взаимной информации:

$$\begin{aligned}
 \sum_i I(M : y_i) &= \sum_i H(y_i) - H(y_i | M) = & (H(y_i) = 1) \\
 &= \sum_i 1 - \mathbb{E}_m (H(y_i | M = m)) = & (y \text{ не зависит от } x, M \text{ определяется } y) \\
 &= \sum_i 1 - \mathbb{E}_m (H(y_i | M = m, x = i)) = \\
 &= \sum_i 1 - H(\mathbb{E}_m(r_m^i)) \\
 &= n - n \sum_i \frac{1}{n} H(\mathbb{E}_m(r_m^i)) \geq & (\text{Неравенство Йенсена}) \\
 &\geq n \cdot (1 - H(\mathbb{E}_{m,i}(r_m^i))) \geq & (\text{Монотонность энтропии на } (0, \frac{1}{2})) \\
 &\geq n \cdot (1 - H(\frac{1}{2} - \delta)) = & (\text{Ряд Тейлора для энтропии}) \\
 &= \Omega(\delta^2 n)
 \end{aligned}$$

Здесь r_m^i — характеристическая функция ошибки $M = m, x = i$.

Пояснение к переходу к r_m^i : Энтропии равны, так как распределения совпадают. Если $\Pr[y_i = 1 | M = m, x = i] = p$, то Алиса ошибется или не ошибется с вероятностью p (так как алгоритм детерминированный).

Чтобы алгоритм был корректен, $\mathbb{E}_{i,m}(r_m^i) \leq \frac{1}{2} - \delta$.

Теперь $\log |M| \geq H(M) \geq \Omega(\delta^2 n)$. То есть нам нужно хотя бы такое количество бит передать. С другой стороны можем построить простой протокол с $2\delta n$ битами. Для этого Боб передаст первые $2\delta n$ бит, а затем Алиса, если нужный ей бит есть, выдаст его, иначе выдаст 0. Вероятность ошибки в таком случае ровно такая, как мы хотели: $\frac{1}{2}(1 - 2\delta)$. Эта верхняя оценка больше нашей нижней. На самом деле она улучшается.

Определение 12

Пусть μ — вероятностная мера на $X \times Y$.

$\text{IC}_\mu^{\text{ext}} := I(\pi(X, Y) : (X, Y))$ — внешнее информационное разглашение.

$\text{IC}_\mu^{\text{int}} := I(\pi(X, Y) : X | Y) + I(\pi(X, Y) : Y | X)$ — внутреннее информационное разглашение.

Интуиция определения следующая. Внешнее информационное разглашение есть количество информации, которое получит внешний наблюдатель об (X, Y) , зная значение протокола $\pi(X, Y)$. Внутреннее информационное разглашение — сколько информации получают внутренние наблюдатели о входах друг друга, посмотрев значения протокола.

Теорема 4.6.1. $D(\pi) \geq \text{IC}_\mu^{\text{ext}}(\pi) \geq \text{IC}_\mu^{\text{int}}(\pi)$

□ Первое неравенство очевидно, так как $I(\pi(X, Y) : (X, Y)) \leq H(\pi(X, Y)) \leq \log |M| \leq D(\pi(X, Y))$. M — листья протокола. Второе докажем **потом**. ■

Теорема 4.6.2 (Храпченко). $L(\text{XOR}) \geq \Omega(n^2)$

□ Покажем, что для любого протокола π задачи KW_{\oplus_n} существует такое распределение μ , что $\text{IC}_\mu^{\text{int}}(\pi) \geq 2 \log n$. Отсюда будет следовать, что $D(\pi) \geq 2 \log n$ и $L(\oplus_n) \geq n^2$. Распределение μ будет равномерным на всех парах вида $(x, x \oplus e_i)$, где $\oplus_n(x) = 1$, то есть в x нечётное число единиц, а строка e_i имеет единицу в позиции i и нули во всех остальных. Таким образом, пары входов из распределения μ всегда будут

отличаться только в одном бите. По определению,

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

Рассмотрим одно из слагаемых $I(\pi : Y | X)$. Имеем:

$$I(\pi : Y | X) = I(\pi : X \oplus e_i | X) = H(X \oplus e_i | X) - H(X \oplus e_i | X, \pi) = H(e_i) = \log n$$

Аналогичное равенство верно и для $I(\pi : X | Y)$. Таким образом, $IC_{\mu}^{int}(\pi) \geq 2 \log n$, что и требовалось. Верхнюю оценку на $D(\pi)$ нетрудно доказать, явно построив функцию. ■

□ Докажем вторую часть [теоремы 4.6.1](#). Для этого докажем $I(\pi : x, y) \geq I(\pi : x | y) + I(\pi : y | x)$.

Если оставить только одно слагаемое, слева останется $H(\pi) - H(\pi | x, y)$, а справа $H(\pi | y) - H(\pi | x, y)$, которое точно не больше.

Пусть π_i – префикс π , то есть то, что Алиса и Боб отправили за i -ый раунд (i -ый бит).

$$I(\pi : x, y) = \sum_i I(\pi_i : x, y | \pi_{<i}) \quad (\text{Chain rule})$$

Аналогично попробуем нарезать слагаемые правой части:

$$I(\pi : x | y) = \sum_i I(\pi_i : x | y, \pi_{<i})$$

$$I(\pi : y | x) = \sum_i I(\pi_i : y | x, \pi_{<i})$$

Вход Боба первое слагаемое «равно нулю»¹, так как π_i определяется y -ом. Аналогично «второе равно» нулю, когда ходит Алиса. Поэтому каждый раз неравенство сохраняется.

Чтобы исправить скрытую фундаментальную ошибку распишем через матожидания

$$I(\pi : x, y) = \sum_i \mathbb{E}_m I(\pi_i : x, y | \pi_{<i} = m)$$

$$I(\pi : x | y) = \sum_i \mathbb{E}_m I(\pi_i : x | y, \pi_i = m)$$

$$I(\pi : y | x) = \sum_i \mathbb{E}_m I(\pi_i : y | x, \pi_i = m)$$

Это уже корректное утверждение, так как для каждого конкретного m одно из слагаемых действительно обнуляется. ■

¹Это и есть баг, на самом деле это случайная величина

Chapter 5

Колмогоровская сложность

5.1 Определения

Пусть F — вычислимая декодирующая функция. Определим $K_F(x) := \min\{|p| \mid F(p) = x\}$. Это «критерий сжимаемости» функции F .

Будем считать, что $F \leq G$ (F не хуже G), если $\forall x K_F(x) \leq K_G(x) + c_{FG}$.

Назовем способ описания (функцию) оптимальным, если она не хуже всех остальных.

Теорема 5.1.1. Существует оптимальный способ описания.

Определение 13

Колмогоровская сложность для x — $K(x) := K_U(x)$, где U — оптимальный способ описания.

Свойства.

1. $K(x) \leq |x| + c$, так как можно взять $K_{id}(x) = |x|$
2. $K(XX) \leq |x| + c$, можно взять описание $F(p) = pp$
3. Пусть Gx не более p единиц, тогда $K(x) \leq H(p) \cdot |x| + c$.
Можем взять $F(p) = p$ -ое слово с не более p единицами.

$$\sum_{i=0}^n \binom{i}{n} \leq 2^{H(p)n}.$$

Теорема 5.1.2. Пусть M — всюду вычислимая функция. Если $M(x) \leq K(x)$ и $\forall c \exists x: M(x) \geq c$, то M не вычислима.

□ Зафиксируем c . Найдем x_c — первое слово, где $M(x_c) \geq c$. Так как M вычислима всюду, определено $F(c)$. Тогда из $F(c) = x_c$ следует, что $K(x) \leq \log c + c_0$ по определению. ■

Следствие 1. У почти всех слов колмогоровская сложность равна $n - \text{const}$.

5.2 Применение

Мы лишаемся вычислимости, поэтому мы лишаемся практических применений, как с энтропией. Но есть математическое применение, так как это оптимальный алгоритм.

Можно доказать, что одноленточная машина Тьюринга, копирующая вход будет работать $\Omega(|x|^2)$.

5.3 Условная сложность

$$K(x | y) = K_U(x | y).$$

Это способ описания, который может еще использовать y : $K_F(x | y) := \min\{|p| \mid F(p, y) = x\}$. Аналогично U — оптимальный способ описания¹.

Замечание. $K(x) = K(x | \emptyset) + \text{const}$

$$K(x, y) := K(\langle x, y \rangle) \quad \langle \cdot, \cdot \rangle — \text{какая-то кодировка пары.}$$

Свойства.

1. $K(x | y) \leq K(x) + \mathcal{O}(1)$
2. $K(x, y) \leq K(x) + K(y | x)$

Теорема 5.3.1 (Колмогоров, Левин). $K(x, y) = K(x) + K(y | x)$

□ В одну сторону по свойству. Пусть $n = K(x, y)$ Пусть $S := \{(a, b) \mid K(a, b) \leq n\}$,

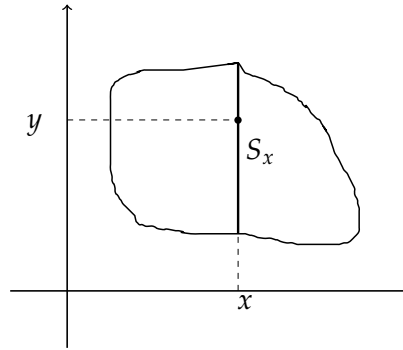


Figure 5.1: kolmo thm

$$K(y | x) \leq \underbrace{\log |S_x|}_m + \mathcal{O}(\log n).$$

Рассмотрим все x , для которых $\log |S_x| \geq m$. Таких не более 2^{n-m+c} , так как мы знаем, что $K(x, y) = n$, то внутри множества размером 2^n , есть элемент с большой сложностью, теперь множество S по размеру не более 2^{n+c} , но так как в одном сечении не более 2^m , получаем 2^{n-m+c} .

Тогда $K(x) \leq n - m$. Если мы подставим в неравенство выше, то получим то, что хотели. ■ Конец.

¹Упражнении — аналогично доказать существования