Дискретная математика 2 семестр ПИ, Лекции

Собрано 9 марта 2022 г. в 17:13

Содержание

Раздел #1: Кодирование информации

1.1. Задача об оптимальном префиксном коде

Пусть Λ – произвольное конечное множество (алфавит), $a \in \Lambda$ – символы. Пусть $\forall a \in \Lambda\ \exists l(a) \in \mathbb{N}, \exists c(a) = \{0,1\}^{l(a)}$ – кодовая последовательность a, где $l(a)$ – длина.
Очевидно, условие $\forall a, b \in \Lambda \to (a \neq b \Rightarrow c(a) \neq c(b))$ не является достаточным для однозначного распознавания символов.

Def 1.1.1. *Код называется префиксным, если* $\forall a, b \in \Lambda\ c(a) = \omega \Rightarrow \nexists\ m \in \mathbb{N}_0 : c(b) = \omega\gamma$, *где* $\gamma \in \{0,1\}^m$

Пусть $\forall a \in \Lambda$ соответствует вероятность $p(a)$ появления этого символа в сообщении. $\sum_{a\in\Lambda} p(a) = 1$ и считаем $\forall a \in \Lambda\ p(a) > 0$.
Введем дискретную случайную величину $l : \forall a \in \Lambda\ Pr\{l = l(a)\} = p(a)$ – длина кодовой последовательности символа в сообщении.

Def 1.1.2. *Оптимальным называется префиксный код, минимизирующий математическое ожидание* $l : \mathbb{E}l = \sum_{a\in\Lambda} l(a)p(a)$

Чем чаще встречается символ, тем короче должна быть кодовая последовательность.
Почему вообще ОПК существует? Известно, что $\mathbb{E}l \geqslant 1$ (в каждой кодовой последовательности должен быть хотя бы один символ). Всегда можно сделать префиксный код, в котором все символы имеют одинаковые длины кодовых последовательностей и эти последовательности различны ($\forall a \in \Lambda\ l(a) = \lceil \log_2(|\Lambda|) \rceil$), т.е. префиксный код существует и матожидание длины кодовой последовательности ограничено.

Lm 1.1.3. Если в некотором коде C существует $x \in \Lambda : c(x) = \omega\alpha$, где $\alpha \in \{0,1\}$ и при этом $\nexists\ y \in \Lambda, y \neq x : c(y) = \omega\gamma$, где $\gamma \in \{0,1\}^k$ (то есть, если ω не является началом никакой другой кодовой последовательности, кроме $c(x)$), то код $C' : c'(x) = \omega, \forall y \in \Lambda, y \neq x\ c' = c(y)$ будет префиксным (по построению и условию леммы) и $\mathbb{E}l' = \mathbb{E}l - p(x)l(x) + p(x)(l(x)-1) = \mathbb{E}l - p(x) < \mathbb{E}l$. Тогда код C точно не мог быть оптимальным.

Lm 1.1.4 (Лемма о кратчайшем префиксе)**.** Если в префиксном коде C $\exists a, b \in \Lambda, a \neq b : p(a) < p(b), l(a) < l(b)$, то такой код не оптимален.

Доказательство. Проверим, что для кода C', в котором $c'(a) = c(b), c'(b) = c(a)$ и $\forall x \in \Lambda : x \neq a, x \neq b\ c'(x) = c(x)$ верно $\mathbb{E}l - \mathbb{E}l' > 0$.

$$\mathbb{E}l - \mathbb{E}l' = p(a)l(a) + p(b)l(b) - p(a)l(b) - p(b)l(a) = (p(a) - p(b))(l(a) - l(b)) > 0$$

■

Lm 1.1.5 (Лемма о соседстве самых редких символов)**.** Пусть $a, b \in \Lambda, a \neq b$ – символы с наименьшими вероятностями ($\forall x \in \Lambda\ p(x) \geqslant p(b) \geqslant p(a)$). Тогда $\exists$ ОПК $: c(a) = \omega 0, c(b) = \omega 1$, где $\exists k \in \mathbb{N}_0 : \omega \in \{0,1\}^k$ и это самые длинные кодовые последовательности.

Доказательство. Пусть C' – ОПК. По лемме о кратчайшем префиксе a и b имеют самые длинные кодовые последовательности в $C' : \forall x \in \Lambda, x \neq a, x \neq b \; l'(a) \geqslant l'(b) \geqslant l'(x)$
Если $c(a) = \omega\gamma, \omega \in \{0,1\}^{l'(b)}, \gamma \in \{0,1\}^{l'(a)-l'(b)}$ и ω не является началом никакой кодовой последовательности (т.к. остальные кодовые последовательности не длиннее ω и $\nexists$ символа с кодовой последовательностью ω в силу префиксности C') $\Rightarrow$ можно сократить кодовую последовательность a, создав более оптимальный код (?!).
$\Rightarrow$ из оптимальности C' следует $l(a) = l(b)$. Пусть $c'(b) = \omega 1$, тогда, если $\exists x \in \Lambda : c'(x) = \omega 0$, то построим ОПК $C : c(a) = c'(x), c(x) = c'(a), \forall z \in \Lambda, z \neq a, z \neq x \; c(z) = c'(z)$.
Если $\nexists \, x \in \Lambda : c'(x) = \omega 0$, то построим ОПК $C : c(a) = \omega 0, \forall z \in \Lambda, z \neq a \; c(z) = c'(z)$. ■

Lm 1.1.6 (Лемма об ОПК для расширенного алфавита)**.** Пусть $a, b \in \Lambda, a \neq b$ – символы с намиеньшими вероятностями. $\Lambda' = \Lambda \setminus \{a,b\} \cup \{\underbrace{ab}\}$, где $\underbrace{ab} \notin \Lambda$, $p(\underbrace{ab}) = p(a)+p(b)$. Пусть C' – ОПК для $\Lambda', c'(\underbrace{ab}) = \omega$. Тогда для Λ код $C : c(a) = \omega 0, c(b) = \omega 1, \forall x \in \Lambda, x \neq a, x \neq b \; c(x) = c'(x)$ будет ОПК.

Доказательство. $l(a)p(a)+l(b)p(b) = (l'(\underbrace{ab})+1)(p(a)+p(b)) = l'(\underbrace{ab})p(\underbrace{ab})+p(\underbrace{ab})$. Тогда $\mathbb{E}l = \mathbb{E}l' + p(\underbrace{ab})$.

Пусть $\overline{C}$ – ОПК для Λ и $\mathbb{E}\bar{l} < \mathbb{E}l$. По лемме о соседстве: $\bar{c}(a) = \gamma 0, \bar{c}(b) = \gamma 1$. Построим $\overline{C}'$ для $\Lambda' : \bar{c}'(\underbrace{ab}) = \gamma$ и $\forall x \in \Lambda, x \neq a, x \neq b \; \bar{c}'(x) = \bar{c}(x)$.

$\overline{C}'$ – префиксный? По Лемме о кратчайшем префиксе $\nexists$ символа с кодовой последовательностью длины $> \bar{l}(a)$. Никакой символ не мог иметь кодовую последовательность γ, т.к. $\overline{C}$ префиксный. Единственные две последовательности длины $\bar{l}(a)$, начинающиеся на γ – это коды a и b. Но их нет в Λ'. При этом $\mathbb{E}\bar{l} = \mathbb{E}\bar{l}' = p(\underbrace{ab})$. По предположению $\mathbb{E}l' + p(\underbrace{ab}) = \mathbb{E}l > \mathbb{E}\bar{l} = \mathbb{E}\bar{l}' + p(\underbrace{ab})$ (?!) оптимальности $C' \Rightarrow \mathbb{E}\bar{l} \geqslant \mathbb{E}l$, но т.к. $\overline{C}$ – ОПК $\Rightarrow \mathbb{E}\bar{l} = \mathbb{E}l$ и C – ОПК. ■

Задача: нужно построить ОПК на алфавите $\Lambda, |\Lambda| = M$. По лемме об ОПК для расширенного алфавита задачу построения ОПК можно свести к такой же задаче, но с исходным алфавитом с числом букв на единицу меньше, и с набором вероятностей, получющимся из первоначального сложением двух наименьших вероятностей.
Уменьшаем пока не получится алфавит из двух букв. ОПК для алфавита из 2-х букв – $\{0,1\}$.
Строже: $\Lambda_0 := \Lambda$. $\forall k \in 0...(M-3)$ берем $a_k, b_k \in \Lambda_k : \forall x \in \Lambda_k, x \neq a_k, x \neq b_k \; p(a_k) \leqslant p(b_k) \leqslant p(x)$ и построим $\Lambda_{k+1} = \Lambda_k \setminus \{a_k, b_k\} \cup \{\underbrace{a_k b_k}\}$...

Для $\Lambda_{M-2} = \{a_{M-2}, b_{M-2}\}$ оптимальным будет код $C_{M-2} : c_{M-2}(a_{M-2}) = 0, c_{M-2}(b_{M-2}) = 1$, т.к. для него $\mathbb{E}l_{M-2} = 1$.
Теперь для $k \in 1...(M-2)$ есть ОПК C_k для Λ_k. По лемме об ОПК для расширенного алфавита строится ОПК C_{k-1} для Λ_{k-1} такой, что $c_{k-1}(a_{k-1}) = c_k(a_{k-1}b_{k-1})0, c_{k-1}(b_{k-1}) = c_k(a_{k-1}b_{k-1})1, \forall x \in \Lambda_k, x \neq \underbrace{a_{k-1}b_{k-1}} \; c_{k-1}(x) = c_k(x)$.

Выполняем, пока не получится C_0 – ОПК для $\Lambda_0 = \Lambda$.

Пример 1.1.7. $\Lambda_0 = \{a,b,c,d,e,f,g\}, p(a) = 0.13, p(b) = 0.08, p(c) = 0.25, p(d) = 0.18, p(e) = 0.03, p(f) = 0.12, p(g) = 0.21$.
$a_0, = e, b_0 = b, \Lambda_1 = \{a, \underbrace{e,b}, c, d, f, g\}, p(a) = 0.13, p(\underbrace{eb}) = 0.11, p(c) = 0.25, p(d) = 0.18, p(f) =$

$0.12, p(g) = 0.21$.
$a_1 = \underbrace{eb}, b_1 = f, \Lambda_2 = \{a, \underbrace{ebf}, c, d, g\}, p(a) = 0.13, p(\underbrace{ebf}) = 0.23, p(c) = 0.25, p(d) = 0.18, p(g) = 0.21$.
$a_2 = a, b_2 = d, \Lambda_3 = \{\underbrace{ad}, \underbrace{ebf}, c, g\}, p(\underbrace{ad}) = 0.31, p(\underbrace{ebf}) = 0.23, p(c) = 0.25, p(g) = 0.21$.
$a_3 = g, b_3 = \underbrace{ebf}, \Lambda_4 = \{\underbrace{ad}, \underbrace{gebf}, c\}, p(\underbrace{ad}) = 0.31, p(\underbrace{gebf}) = 0.44, p(c) = 0.25$. $a_4 = c, b_4 = \underbrace{ad}, \Lambda_5\{\underbrace{cad}, \underbrace{gebf}\}, p(\underbrace{cad}) = 0.56, p(\underbrace{gebf}) = 0.44$. Тогда $c_5(\underbrace{gebf}) = 0, c(\underbrace{cad}) = 1$.
Теперь раскрываем алфавит обратно:
$c_4(\underbrace{gebf}) = 0, c_4(c) = 10, c_4(\underbrace{ad}) = 11$.
$c_3(g) = 00, c_3(\underbrace{ebf}) = 01, c_3(c) = 10, c_3(\underbrace{ad}) = 11$.
$c_1(g) = 00, c_1(\underbrace{eb}) = 010, c_1(f) = 011, c_1(c) = 10, c_1(a) = 110, c_1(d) = 111$.
$c_0(g) = 00, c_0(e) = 0100, c_0(b) = 0101, c_0(f) = 011, c_0(c) = 10, c_0(a) = 110, c_0(d) = 111$.

1.2. Неравенство Крафта

Пусть задан набор длин $l_1, ..., l_m$, не все обязательно различны. Может ли такой набор оказаться набором длин некоторого префиксного кода.

Теорема 1.2.1. Для того, чтобы набор длин $l_1, ..., l_m$ мог быть набором длин кодовых последовательностей некоторого ПК для алфавита из m символов необходимо и достаточно, чтобы $\sum_{i=1}^{m} 2^{-l_i} \leqslant 1$.

Доказательство. " $\Rightarrow$ ". Пусть $\exists$ префиксный код для алфавита с кодовыми последовательностями с длинами $l_1, ..., l_m$. множество кодовых последовательностей – набор всех путей на двоичном дереве от корня к листьям.
Корень – нулевой уровень. Далее последовательно увеличиваем номер по мере удаления от корня.
Каждой вершине v на уровне t сопоставим число $a(v) = 2^{-t}$
Пусть вершина v на уровне t – не лист. Т.е. на уровне $t+1$ есть $\geqslant 1$ вершина, получившаяся из v. Обозначим её $N(v)$. Тогда $a(v) \geqslant \sum\limits_{u \in N(v)} a(u)$
Просуммируем неравенства для всех не листов:

$$\sum_{v \text{ не лист}} a(v) \geqslant \sum_{u \text{ не корень}} a(u)$$

$\Rightarrow 2^0 \geqslant \sum\limits_{u \text{ листья}} a(u)$. Необходимость доказана.
" $\Leftarrow$ ". Считаем, что выполнено неравенство и пусть $l_1 \leqslant l_2 \leqslant ... \leqslant l_m$
n_j– число листьев на уровне $j : n_j = |\{i : l_i = j, i \in 1 : m\}|$

$\sum\limits_{i \in 1:m} 2^{-l_i} \geqslant 1 \Rightarrow \sum\limits_{j \in 1:l_m} 2^{-j} n_j \leqslant$. Тогда $\forall j \in 1 : l_m : n_j \leqslant 2^j - (2^{j-1} n_1 + ... + 2 n_{j-1})$
Пусть $m \neq 1$. Выделим на первом уровне вершин $n_1 \leqslant 2$, на втором уровне останется $2(2 - n_1)$. Известно, что $n_2 \leqslant 2^2 - 2n_1 \Rightarrow$ осталось не меньше, чем требуется для второго уровня.

$(j-1)$–уровень: было свободно $2^{j-1} - (2^{j-2}n_1 + ... + 2n_{j-2})$ и n_{j-1} не больше этой величины. Выделим n_{j-1} узлов, останется $2^{j-1} - (2^{j-2}n_1 + ... + 2n_{j-2}) - n_{j-1}$. Значит на j-м уровне будет $2 \cdot (...) = 2^j - (2^{j-1}n_1 + ... + 2n_{j-1})$ ■

1.3. Напоминалка

Пусть S – конечное множество. $|S| = n$.
Пусть задана функция $f : S \to [0,1], \forall \omega \in S \exists! f(\omega) \in [0,1]$
$\sum\limits_{\omega \in S} f(\omega) = 1$. Определим $\forall A \subseteq S$ величину $Pr(A) = \sum\limits_{\omega \in A} f(\omega)$
Функция f в общем-то и не нужна. Достаточно иметь Pr.

Def 1.3.1. *(S, Pr) называется вероятностным пространством.*
S – пространство элементарных событий.
$\omega \in S$ – элементарное событие (исход). $A \subseteq S$ – событие. $Pr(A)$ – вероятность A.
$A, B \subseteq S, Pr(A \cap B) = 0$ – несовместные события.

Свойства вероятности:

- $Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$
- $Pr(A) + Pr(S \backslash A) = 1$
- $Pr(A \cup B) \leqslant Pr(A) + Pr(B)$
- $Pr(A) = Pr(A \backslash B) + Pr(A \cap B)$

Неравенство Йенсена:

Def 1.3.2. *Функция f называется выпуклой на $X \in R$, если $\forall x_1, x_2 \in X$ и $\forall \alpha \in [0,1]$ выполняется неравенство $f(\alpha x_1 + (1 - \alpha x_2) \leqslant \alpha f(x_1) + (1-\alpha) f(x_2)$*

Неравенство Йенсена: пусть f выпуклая на X функция. Тогда $f(\sum\limits_{i=1}^{n} \alpha_i x_i) \leqslant \sum\limits_{i=1}^{n} \alpha_i f(x_i)$, где
$x_i \in X, \alpha_i \geqslant 0, \sum\limits_{i=1}^{n} \alpha_i = 1$.

Доказательство. База при n = 2 верна по определению выпуклой функции. Пусть f – выпуклая на X функция. Тогда
$f(\sum\limits_{i=1}^{n+1} \alpha_i x_i) = f((1-\alpha_{n+1}) \sum\limits_{i=1}^{n} \frac{\alpha_i}{1-\alpha_{n+1}} x_i + \alpha_{n+1} x_{n+1}) \leqslant (1-\alpha_{n+1}) f(\sum\limits_{i=1}^{n} \frac{\alpha_i}{1-\alpha_{n+1}} x_i) + \alpha_{n+1} f(x_{n+1}) \leqslant$
$\leqslant (1-\alpha_{n+1}) \sum\limits_{i=1}^{n} \frac{\alpha_i}{1-\alpha_{n+1}} f(x_i) + \alpha_{n+1} f(x_{n+1}) = \sum\limits_{i=1}^{n+1} \alpha_i f(x_i)$ ■

1.4. Конечная случайная схема

Def 1.4.1. *Пусть $A_1, A_2, ..., A_n$– разбиение множества исходов S вероятностного пространства (S, Pr). Конечной случайной схемой называется схема α, сопоставляющая каждому A_i вероятность $Pr(A_i)$*

Def 1.4.2. *Энтропией КСС называется* $H(\alpha) = -\sum_{i=1}^{n} Pr(A_i) \times \log Pr(A_i)$

Свойства энтропии:

- $H(\alpha) \geqslant 0$
- Энтропия характеризует неопределенность, заключенную в КСС
- Для любой $\alpha \subset k$ исходами справедливо $H(\alpha) \leqslant \log k$

Доказательство. $f(x) := -x \cdot \log x$. На $[0,1]$ функция $f(x)$ строго вогнутая $\Rightarrow$ по неравенству Йенсена $\sum_{i=1}^{n} \lambda_i \cdot f(x_i) \leqslant f(\sum_{i=1}^{n} \lambda_i \cdot x_i)$, причём равенство $\Leftrightarrow x_1 = ... = x_n$.

Тогда возьмём $x_i = Pr(A_i)$ и $\lambda_i = \frac{1}{k}$ $\forall i \in 1...k$, получаем $\sum_{i=1}^{k} \frac{1}{k}(-Pr(A_i) \times \log Pr(A_i)) \leqslant$

$\leqslant -\sum_{i=1}^{k} \frac{1}{k} Pr(A_i) \times \log(\sum_{i=1}^{k} Pr(A_i))$

$-\frac{1}{k} \sum_{i=1}^{k} Pr(A_i) \times \log Pr(A_i) \leqslant -\frac{1}{k} \log \frac{1}{k}$

$-\sum_{i=1}^{k} Pr(A_i) \times \log Pr(A_i) \leqslant \log k$

Максимальная энтропия для КСС имеет схема с k равновероятностными исходами.
$H(\alpha) = 0 \Leftrightarrow \exists!$ достоверный исход в α ■

Пусть есть КСС α с исходами $A_1, ..., A_k$ и КСС β с исходами $B_1, ..., B_l$. Их пересечением $\alpha \cap \beta$ называются КСС, исходы которой – $A_i \cap B_j, \forall i \in 1, ..., k, j \in 1, ..., l$

Тогда $H(\alpha \cap \beta) = -\sum_{i=1}^{k} \sum_{j=1}^{l} Pr(A_i \cap B_j) \times \log Pr(A_i \cap B_j)$

Т.к. $Pr(A_i \cap B_j) = Pr(A_i) \times Pr(B_j|A_i) \Rightarrow H(\alpha \cap \beta) =$

$= -\sum_{i=1}^{k} \sum_{j=1}^{l} Pr(A_i) Pr(B_j|A_i) \times (\log Pr(A_i) + \log Pr(B_j|A_i)) =$

$= -\sum_{i=1}^{k} \sum_{j=1}^{l} Pr(A_i) Pr(B_j|A_i) \times \log Pr(A_i) - \sum_{i=1}^{k} \sum_{j=1}^{l} Pr(A_i) Pr(B_j|A_i) \times \log Pr(B_j|A_i) =$

$= -\sum_{i=1}^{k} Pr(A_i) \cdot \log Pr(A_i) \cdot \sum_{j=1}^{l} Pr(B_j|A_i) + \sum_{i=1}^{k} Pr(A_i) \cdot (-\sum_{j=1}^{l} Pr(B_j|A_i) \cdot \log Pr(B_j|A_i)) =$

$= -\sum_{i=1}^{k} Pr(A_i) \cdot \log Pr(A_i) + ... = H(\alpha) + ...$

Def 1.4.3. *Величину* $H(\beta|A_i) := -\sum_{j=1}^{l} Pr(A_i) Pr(B_j|A_i) \cdot \log Pr(B_j|A_i)$ *называют условной энтропией* β *при условии* A_i

Def 1.4.4. *Величину* $H_\alpha(\beta) := \sum_{i=1}^{k} Pr(A_i) \cdot H(\beta|A_i)$ *называют средней условной энтропией* β *при условии* α.

Таким образом, $H(\alpha \cap \beta) = H(\alpha) + H_\alpha(\beta)$
Докажем, что $0 \leqslant H_\alpha(\beta) \leqslant H(\beta)$

Неотрицательность следует из неотрицательности энтропий.
fix j, $f(x) = -x \cdot \log x, \lambda_i = Pr(A_i), x_i = Pr(B_j|A_i) \quad \forall i \in 1...k$

Неравенство Йенсена: $\sum\limits_{i=1}^{k} Pr(A_i) \cdot (-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leqslant$

$\leqslant (-\sum\limits_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i)) \cdot \log \sum\limits_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i)$

ПЧ $= (-\sum\limits_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i)) \cdot \log \sum\limits_{i=1}^{k} Pr(A_i) \cdot Pr(B_j|A_i) =$

$= -(\sum\limits_{i=1}^{k} Pr(B_j \cap A_i)) \cdot \log \sum\limits_{i=1}^{k} Pr(B_j \cap A_i) = -Pr(B_j) \cdot \log Pr(B_j)$

Просуммируем по j: $\sum\limits_{j=1}^{j} \sum\limits_{i=1}^{k} Pr(A_i) \cdot (-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leqslant \sum\limits_{j=1}^{l} (-Pr(B_j) \cdot \log Pr(B_j))$

$\sum\limits_{i=1}^{k} Pr(A_i) \cdot \sum\limits_{j=1}^{l} (-Pr(B_j|A_i) \cdot \log Pr(A_i)) \leqslant -\sum\limits_{j=1}^{l} Pr(B_j) \cdot \log Pr(B_j)$

$\sum\limits_{i=1}^{k} Pr(A_i) \cdot H(\beta|A_i) \leqslant H(\beta) \Rightarrow H_\alpha(\beta) \leqslant H(\beta)$

$H_\alpha(\beta) = H(\beta) \Leftrightarrow$ все $Pr(B_j|A_i)$ равны между собой.

Формула полной вероятности: $\forall j \in 1...l Pr(B_j) = \sum\limits_{i=1}^{k} Pr(B_j|A_i) \cdot Pr(A_i)$

$\forall j \in 1...l Pr(B_j) = Pr(B_j|A_1) \cdot \sum\limits_{i=1}^{k} Pr(A_i) = Pr(B_j|A_1).$

То есть $\forall i \in 1...k, j \in 1...l \quad Pr(B_j) = Pr(B_j|A_i)$

Def 1.4.5. *События A и B – взаимно независимы* $\Leftrightarrow Pr(A \cap B) = Pr(A) \cdot Pr(B) \Leftrightarrow Pr(A) \cdot Pr(B|A) = Pr(A) \cdot Pr(B) \Leftrightarrow Pr(B|A) = Pr(B)$

Def 1.4.6. *КСС α и β называются независимыми, когда все исходы α независимы со всеми исходами β. В таком случае $H_\alpha(\beta)$ максимальна и равна $H(\beta)$*

1.5. Количество информации

Def 1.5.1. *Величина $I(\alpha, \beta) = H(\beta) - H_\alpha(\beta)$ называется количеством информации.*

Свойства:

1. $I(\alpha, \beta) \geqslant 0$
2. $I(\alpha, \beta) = H(\beta) \Leftrightarrow H_\alpha(\beta) = 0$
3. $I(\alpha, \beta) = I(\beta, \alpha)$
4. $I(\alpha, \beta) = 0 \Leftrightarrow \alpha$ и β независимы.

Пример 1.5.2. Загадано натуральное число $x \in 1...N$
β– опыт, состоящий в нахождении x, β_m – опыт, показывающий, делится ли x на m, $m \in 1...N$.
У β есть N исходов, у β_m – два исхода.
$H_{\beta_m}(\beta) = Pr(x \vdots m) \cdot H(\beta|x \vdots m) + Pr(x \nmid m) \cdot H(\beta|x \nmid m)$

$q := \left|\frac{N}{m}\right|$ – количество чисел от 1 до N, делящихся на m. Тогда $Pr(x \vdots m) = \frac{q}{N}, Pr(x \nmid m) = \frac{N-q}{N}$.
$H(\beta|x \vdots m) = -\sum\limits_{i \vdots m, i \in 1...m} \frac{1}{q} \cdot \log \frac{1}{q} = -\frac{q}{q} \cdot \log \frac{1}{q} = \log q.$

Аналогично $H(\beta|x \nmid m) = \log(N-q) \Rightarrow H_{\beta_m}(\beta) = \frac{q}{N} \cdot \log q + \frac{N-q}{N} \cdot \log(N-q)$

$I(\beta_m, \beta) = \log N - \frac{q}{N} \cdot \log q - \frac{N-q}{N} \cdot \log(N-q) =$
$= \frac{q}{N} \cdot \log N - \frac{q}{N} \log q - \frac{N-q}{N} \cdot \log N - \frac{N-q}{N} \cdot \log(N-q) =$
$= -\frac{q}{N} \cdot \log \frac{q}{N} - \frac{N-q}{N} \cdot \log \frac{N-q}{N} \leqslant \log 2$
Равенство достигается при $q = N - q = \frac{N}{2}$.

Пример 1.5.3 (Данетки)**.** Загадано число от 1 до N.
Опыт β – угадать число.
Опыт α – задать любой общий (да/нет) вопрос и получить ответ.
$H(\beta) = \log N$ (Числа загаданы с равной вероятностью)
$H(\alpha) \leqslant \log 2$ (Поскольку есть всего 2 варианта ответа)
$H(\alpha_1\alpha_2...\alpha_k) \leqslant \log 2^k = k \log 2$ (k вопросов, 2 варианта ответа)
Чтобы угадать число потребуется $k \geqslant \frac{\log N}{\log 2} = \log_2 N$ вопросов.
Есть ли алгоритм, который умеет угадывать загаданное число за $O(\log N)$.

1.5.1. Избыточное кодирование

Есть сообщение $u \in \{0,1\}^k$, которое нужно передать.
Можем передавать сообщение $x(u) \in \{0,1\}^n, n \geqslant k$, содержащую некоторую избыточную информацию (канал связи шумит и может допускать ошибки), но не более d ошибок на сообщение.
β заключается в нахождении всех d ошибок. Сколько у β исходов? Для каждого количества ошибок j от 0 до d есть $\binom{n}{j}$ вариантов их расположения, то есть всего исходов у β ровно $\sum\limits_{j=0}^{d} \binom{n}{j}$

Следовательно, $H(\beta) = \log \sum\limits_{j=0}^{d} \binom{n}{j}$
α – дополнительное сообщение размера $n-k$. Их 2^{n-k} и $\Rightarrow H(\alpha) = \log 2^{n-k} = (n-k)\log 2$
Чтобы гарантированно найти все ошибки нужно $H(\alpha) \geqslant H(\beta)$

$$(n-k)\log 2 \geqslant \log \sum_{j=0}^{d} \binom{n}{j} \Rightarrow n-k \geqslant \log_2 \sum_{j=0}^{d} \binom{n}{j} \Rightarrow k \leqslant n - \log_2 \sum_{j=0}^{d} \binom{n}{j}$$

Таким образом, если канал связи допускает не более d ошибок, то для передачи сообщения размером k понадобится не менее $k + \log_2 \sum_{j=0}^{d} \binom{n}{j}$.
Или, поскольку количество ошибок обычно зависит от размера переданного сообщения, если передаётся n бит и из них не более d могут быть ошибочными, то в переданном сообщении можно закодировать сообщение длиной не более $n - \log_2 \sum\limits_{j=0}^{d} \binom{n}{j}$.

1.5.2. Код Хэмминга

Предыдущая задача при $d = 1$. Известно, что $2^{n-k} \geqslant \sum_{j=0}^{1} \binom{n}{j} = 1 + n$.

$l := n - k$– длина "избыточного сообщения". Тогда $k \leqslant 2^l - l - 1$.

Чем большее сообщение, тем относительно меньше лишней информации. Как передавать дополнительную информацию?

Пример 1.5.4. Пусть k = 12 и мы хотим передать сообщение u = 101101011100. Зарезервируем в сообщении длины 17 места с номерами $2^i(1, 2, 4, 8, 16)$, а на остальные позиции запишем сообщение:

$x_0(u)$ = __1_011_0101110_0

Подберём на позицию 2^i такую цифру, чтобы произведение $x(u)$ и i-й строки матрицы было равно 0. На "неопределенных"позициях в строке с номером i стоят $0(2^j = 10\ldots0)$.

На позиции 2^i в i–й строке стоит 1.

$?*1+\_*0+1*1+\_*0+0*1+1*0+1*1+\_*0+0*1+1*0+0*1+1*0+1*1+1*0+0*1+\_*0+0*1 =$

$=? + 1 + 1 + 1 = 1+? = 0 \Rightarrow ? = 1.$

Получается x_1 = 1_1_011_0101110_0

Аналогично делаем для остальных. Итого $x(u)$ = 11110110010111000

Как определять позицию ошибки? y = 11110110000111000

Посчитаем $A \times y^T = (0, 1, 0, 1, 0)^T$ – двоичная запись позиции с ошибкой.

Старший бит – справа. Почему так?

При умножении на i -ю строку матрицы j -я позиция сообщения влияла только если $A[i, j] = 1$, то есть если на i-м месте в двоичной записи числа j стояла 1 $\Rightarrow$ результат произведения строки матрицы на столбец сообщения изменился (став 1) только для тех строк, где на 10-й позиции стояла 1 (а это строки с номерами, равными позициям, где в двоичной записи числа 10 стоят 1), а для остальных строк остался 0.

Раздел #2: Графы

Def 2.0.1. *Ориентированным графом называют $G = (V, E)$, где $V \neq \varnothing$ – множество вершин, $E \subseteq V \times V$ – множество ребер. Ребра часто записывают по их концам: (v_1, v_2) или v_1v_2.*

Замечание 2.0.2. $V = \varnothing$ иногда встречается в доказательствах утверждений. Пустым графом называют граф, множество ребер которого пусто.

Def 2.0.3. *Пусть $G = (V, E)$. $G' = (V', E')$ называют подграфом G ($G' \leqslant G$), если $V' \subseteq V, E' \subseteq (V' \times V') \cap E$. Если $E' = (V' \times V') \cap E$ подграф, то подграф называют порожденным. Порожденный граф обозначают $G[V']$.*

Def 2.0.4. *Пусть $G = (V, E)$. Путем называется последовательность вершин $v_0v_1...v_n : \forall i \in 0...n\ v_i \in V, \forall i \in 1...n\ (v_{i-1}, v_i) \in E$. Простым называется путь, в котором все вершины различны.*

Def 2.0.5. *Циклом называется последовательность вершин $v_0v_1...v_n : \forall i \in 0...n\ v_i \in V, \forall i \in 1...n\ (v_{i-1}, v_i) \in E, v_0 = v_n$. Простым называется цикл, в котором все вершины, кроме первой и последней, различны.*

Def 2.0.6. *Ациклическим графом называется орграф без циклов.*

2.1. Отношение достижимости

Def 2.1.1. *На множестве вершин V зададим отношение достижимости R^* : вершина $v_1 \in V$ находится в отношении R^* с вершиной $v_2 \in V$ (в этом случае говорят, что вершина v_2 достижима из вершины v_1), если существует с началом v_1 и концом v_2.*

Замечание 2.1.2. Отношение достижимости для вершин орграфа рефлексивно и транзитивно, но не обязательно симметрично.

Def 2.1.3. *Определим с помощью отношения достижимости разбиение множества вершин графа на классы эквивалентности: вершины v_1, v_2 принадлежат одному классу, если отношение симметрично. Такое отношение рефлексивно, транзитивно и симметрично.*

Замечание 2.1.4. Если граф ациклический, то каждый класс эквивалентности состоит из одной вершины.

2.1.1. Граф-покрытие и граф достижимости

Def 2.1.5. *Минимальный граф G_b, индуцирующий на множестве вершин $V(G)$ то же отношение достижимости, что и исходный орграф G (т.е. граф с неуменьшаемым далее множеством ребер), называется **базисным** графом для графа G.*

Замечание 2.1.6. Базисный граф не обязательно единственный.

Замечание 2.1.7. В конечном орграфе существует базисный граф. Получается последовательным удалением ребер (v_1, v_2), для которых существует не содержащий его путь.

Def 2.1.8. *Классы эквивалентности по отношению достижимости называются связными компонентами. Классы эквивалентности по отношению взаимной достижимости называются компонентами сильной связности.*

Def 2.1.9. *Пусть $G = (V, E)$ – орграф. Граф достижимости (графом транзитивного замыкания) $G^* = (V, E^*)$ для G имеет то же множество вершин V и следующее множество ребер $E^* = \{(u, v) | в\ графе\ G\ вершина\ v\ достижима\ из\ вершины\ u\}$*

Замечание 2.1.10. Ребра графа достижимости G^* соответствуют путям исходного графа G.

Def 2.1.11. *Матрица смежности орграфа $G = (V, E)$ с $|V| = n$ называется матрица A_G размера $n \times n$ с элементами*

$$A_{ij} = \begin{cases} 1, (v_i, v_j) \in E \\ 0 \end{cases}$$

Введем обозначения $\hat{A} := A_G \vee E_n, \hat{A}_0 = E_n, \hat{A}_1 = \hat{A}, ..., \hat{A}_{k+1} = \hat{A}_k \wedge \hat{A}$.

Lm 2.1.12. Пусть $\hat{A}_k = \left(a_{ij}^{(k)}\right)$. Тогда

$$a_{ij}^{(k)} = \begin{cases} 1, \exists \text{ путь из } v_i \text{ в } v_j \text{ длины } \leqslant k, \\ 0 \end{cases}$$

Доказательство. Индукция по k. База верна по определению $\hat{A}_0$. Пусть верно для k, докажем для $k+1$.
$a_{ij}^{(k+1)} = a_{i1}^{(k)} a_{1j}^{(1)} \vee ... \vee a_{ir}^{(k)} a_{rj}^{(1)} \vee ... \vee a_{in}^{(k)} a_{nj}^{(1)}$. Пусть в G из v_i в v_j есть путь длины $\leqslant k+1$. Рассмотрим кратчайший из таких путей. Если длина $\leqslant k$, то $a_{ij}^{(k)} = 1$ и , т.к. $a_{jj}^{(1)} = 1$, то $a_{ij}^{(k)} a_{jj}^{(1)} = 1$ и $a_{ij}^{(k+1)} = 1$.
Если длина ровно $k+1$, то пусть v_r – предпоследняя вершина. Тогда из v_i в v_r есть путь длины k и по предположению $a_{ir}^{(k)} = 1$. Т.к. есть ребро (v_r, v_j), то $a_{ir}^{(k)} a_{rj}^{(1)} = 1$. Поэтому $a_{ir}^{(k)} a_{rj}^{(1)} = 1$ и $a_{ij}^{(k+1)} = 1$.
В другую сторону: пусть $a_{ij}^{(k+1)} = 1$, тогда $\exists r : a_{ir}^{(k)} a_{rj}^{(1)} = 1$. Если это $r = j$, то $a_{ij}^{(k)} = 1$ и по предположению в G есть путь из v_i в v_j длины $\leqslant k$.
Если $r \neq j$, то $a_{ir}^{(k)} = 1$ и $a_{rj}^{(1)} = 1$. Это означает, что в G есть путь из v_i в v_r длины $\leqslant k$ и ребро (v_r, v_j). Объединяем и получаем путь из v_i в v_j длины $\leqslant k+1$. ■

Следствие 2.1.13. Пусть $G = (V, E)$ – орграф, $|V| = n, G^*$ – его граф достижимости. Тогда $A_{G^*} = \hat{A}_{n-1}$.

Замечание 2.1.14. При вычислении можно хитрить: считать $\hat{A} \Rightarrow \hat{A}_2 \Rightarrow \hat{A}_4 \Rightarrow ...$
Также, т.к. на диагонали $\hat{A}$ стоят единицы, то $\forall i < j$ все единицы в $\hat{A}_i$ сохраняются в $\hat{A}_j$ (и в $(\hat{A}_i)^2$). При вычислении квадратов, если в "сумме"обнаруживается $r : a_{ir} = 1$ и $a_{rj} = 1$, то остальные слагаемые можно не рассматривать.

Def 2.1.15. *Граф сильной достижимости $G_*^* = (V, E_*^*)$, где $E_*^* = \{(u, v) | u, v\ взаимно\ достижимы\ в\ G\}$*

По матрице сильной достижимости можно выделить компоненты сильной связности графа G:

1. В первую компоненту K_1 поместить вершину v_1 и все вершины $v_j : A_{G_*^*}(1, j) = 1$

2. Построение $K_1, ..., K_i$ и v_k вершина с минимальным индексом без компоненты. Помещаем её в K_{i+1} и все $v_j : A_{G_*^*}(k, j) = 1$.

Def 2.1.16. *Пусть K и K' – компоненты сильной связности графа G. Компонента K достижима из компоненты K', если $K = K'$ или существуют две такие вершины $u \in K$ и $v \in K'$, что u достижима из v. K строго достижима из K', если $K \neq K'$ и K достижима из K'.*

Def 2.1.17. *Отношение строго достижимости можно представлять в виде орграфа, вершины – компоненты сильной связности, ребра есть если есть строга достижимость – **Конденсация G, ациклический граф**.*

2.2. Турниры и полустепени в орграфе

Def 2.2.1. *Полустепень захода в орграфе для вершины – число дуг, входящих в вершину. Обозначается $d^+(v)$. Полустепень исхода в орграфе для вершины v – число дуг, исходящих из вершины ($d^-(v)$).*

Def 2.2.2. *Турнир – некоторый полный орграф (V, E) (орграф без петель и между любой парой вершин есть ровно одно ребро).*

Def 2.2.3. *Для ребра $(u, v) \in E$ говорим, что u доминирует над v.*

Def 2.2.4. *Турнир (будучи орграфом) транзитивен, если из $(u, v) \in E, (v, w) \in E$ следует из $(u, w) \in E$.*

Def 2.2.5. *Порядком турнира T называется число его вершин.*

Def 2.2.6. *Полустепень выхода вершины v турнира T – число вершин, над которыми v доминирует (еще называется результатом).*

Def 2.2.7. *Последовательность результатов турнира T – упорядоченная последовательность $S = (s_1, ..., s_n)$, где s_i – результат $v_i, 1 \leqslant i \leqslant n$, причем $s_1 \leqslant s_2 \leqslant ... \leqslant s_n$.*

Def 2.2.8. *Множество результатов турнира T – это последовательность $D = (d_1, ..., d_m)$ различных результатов вершин турнира T, где $d_1 < d_2 < ... < d_m$.*

Def 2.2.9. *Если последовательностью результатов турнира T является S, а множество результатов – D, то будем говорить, что S генерирует D.*

Теорема 2.2.10 (Редеи-Камиона для пути)**.** Любой турнир порядка n содержит гамильтонов путь (т.е. путь, содержащий все n вершин).

Упражнение 2.2.11. Доказательство.

Теорема 2.2.12 (Редеи-Камиона для цикла)**.** В сильно связном турнире есть гамильтонов цикл. Верно и обратное утверждение.

Упражнение 2.2.13. Доказательство.

Def 2.2.14. *Вершина $v \in V(T)$ турнира T является королем $\Leftrightarrow \forall x \in V(T)\ \exists$ путь из v в x длиной $\leqslant 2$.*

Теорема 2.2.15. В любом турнире существует вершина-король.

Теорема 2.2.16. Для турнира порядка n следующие утверждения эквивалентны:

1. T транзитивен
2. T не содержит циклов длины 3
3. T ацикличен
4. Последовательность результатов турнира T – это $(0, 1, ..., n-1)$.
5. T содержит ровно один гамильтонов путь.

Доказательство. $1 \Rightarrow 2 : \exists (u,v),(v,w),(w,u)$. Но также $\exists (u,w)$ (?!)
$2 \Rightarrow 3 : \exists$ цикл $(v_1, ..., v_k), k \geqslant 4$. Т.к. нет циклов длины 3, то есть транзитивность. Индукцией покажем, что $\exists (v_1, v_{k-1})$. База: $(v_1, v_2), (v_2, v_3) \in E \Rightarrow (v_1, v_3) \in E$. Переход: $(v_1, v_i) \in E \ \forall i < k-1$, также $(v_i, v_{i+1}) \in E \Rightarrow (v_1, v_{i+1}) \in E$. (?!) цикл (v_1, v_{k-1}, v_k).
$3 \Rightarrow 4 : D^+(T)$ – множество степеней захода. Индукция по n. База очевидна. Переход: пусть верно для $n-1$. В ациклическом графе есть вершина-сток $t : d^+(t) = 0$. Рассмотрим граф $T - t$. $D^+(T-t) = (0, 1, ..., n-2)$. А из $\forall v \in V \setminus t$ ведет одно ребро в t.
$4 \Rightarrow 5$: Существует по теореме Редеи-Камиона. Надо единственность. Докажем по индукции. База очевидна, переход: берем $s : d^-(s) = 0$ (все ребра выходят, исток). Она будет первой в гамильтоновом пути. Рассмотрим $T - s : s$ была соединена со всеми, степени уменьшились на 1 и $D^-(T-s) = (0, 1, ..., n-2)$. Значит в $T - s$ $\exists!$ гамильтонов путь. Если $\exists 2$ гамильтоновых пути с началом в s, то будет и 2 гамильтоновых пути в $T - s$ (?!).
$5 \Rightarrow 1 : P = (v_1, ..., v_n)$ – ! гамильтонов путь. Пусть $\exists m$ – наименьший индекс : в v_m идет ребро из вершины с большим индексом, а v_k – вершина с наибольшим индексом, из которой ребро ведет в v_m.
$m \neq 1, k \neq n$: есть ребро из v_{m-1} в v_{m+1} (минимальность v_m) и из v_m в v_{m+1} (максимальность k). Есть еще цикл $P_1 = (v_1, ..., v_{m-1}, v_{m+1}, ..., v_k, v_m, v_{k+1}, ..., v_n)$.

- $m \neq 1, k = n : P_1 = (v_1, ..., v_{m-1}, v_{m+1}, ..., v_n, v_m)$
- $m = 1, k \neq n : P_1 = (v_2, ..., v_k, v_1, v_{k+1}, ...)$
- $m = 1, k = n : P_1 = (v_2, ..., v_n, v_1)$

Значит такого m не существует и $(v_i, v_j) \in E \Leftrightarrow i < j$. Значит $\forall i, j, k : 1 \leqslant i, j, k \leqslant n$ $(v_i, v_j) \in E$ и $(v_j, v_k) \in E \Rightarrow i < j \vee j < k \Rightarrow (v_i, v_k) \in E$. ■

Теорема 2.2.17. Конденсация любого турнира является транзитивным турниром.

Доказательство. U, V – компоненты сильной связности. $u \in U, v \in V : (u, v) \in E$ или $(v, u) \in E$. Т.е. в конденсации есть либо ребро (U, V), либо (V, U). Рассмотрена произвольная пара вершин конденсации турнира, получилось, что она тоже турнир. Знаем, что конденсация ациклична $\Rightarrow$ транзитивна. ■

Теорема 2.2.18 (Ландау)**.** Некоторая неубывающая последовательность неотрицательных целых чисел $S = (s_1, s_2, ..., s_n)$ является последовательностью результатов некоторого турнира $\Leftrightarrow \sum_{i=1}^{k} s_i \geqslant \frac{k(k-1)}{2}, 1 \leqslant k \leqslant n$, причем равенство при $k = n$.

Замечание 2.2.19. Восстановление турнира по некоторому допустимому множество результатов – это более сложная задача, чем восстановление турнира по некоторой допустимой последовательности результатов.

Теорема 2.2.20 (Яо)**.** Если $m \geqslant 1, D = (d_1, ..., d_m)$ – множество неотрицательных чисел, то существует турнир с множеством результатов D.

Замечание 2.2.21. Теорема Яо доказывает только существование соответствующего турнира, но не дает способ его построения.

Замечание 2.2.22. Проверка существования турнира с заданной последовательностью результатов – линейная задача (Ландау). Построение турниров по последовательности результатов делается быстро (квадратичные алгоритмы).

Замечание 2.2.23. А как построить турнир по множеству? Строят по множеству последовательность (за полиномиальное время), а дальше понятно.