

Дискретная математика

Вдовец Илья, Войко Андрей и Глебский Никита, Блохтин Никита

30 января 2024 г.

Содержание

1 Лекция 13	3
1.1 Вероятность	3
1.2 Свойства вероятности	3
1.3 Оценка объединения	4
1.4 Нижняя оценка на числа Рамсея	4
1.5 Формула включений-исключений в вероятностном смысле	4
2 Лекция 14	5
2.1 Задача о беспорядках.	5
2.2 Условная вероятность	6
2.3 Формулы Байеса и полной вероятности.	6
2.4 Независимые события.	8
3 Лекция 15	8
3.1 Раскраски графов. Хроматическое число графа.	8
3.2 Конструкция Зыкова-Мыцельского графа без треугольников со сколь угодно большим хроматическим числом.	10
3.3 Хроматический многочлен.	11
4 Лекция 16	13
4.1 Математическое ожидание	13
4.2 Неравенство Маркова	13
4.3 Дисперсия	14
4.4 Неравенство Чебышева	14
4.5 Общий случай вероятностного метода	15
5 Лекция 17	15
5.1 Вероятностный метод. Разрезы в графах.	15
5.2 Независимые случайные величины	16
5.3 Оценка биномиальных коэффициентов	17
5.4 Неравенство Чернова	18
6 Лекция 18	19
6.1 Производящие функции	19
6.2 Дифференцирование	20
6.3 Пример применения	21
6.4 Ещё один пример применения (бонусный)	21

7 Лекция 19	22
7.1 Неупорядоченные выборки	22
7.2 Бином Ньютона	22
7.3 Линейные рекуррентные соотношения с постоянными коэффициентами	23
7.4 Теорема о реккуррентном соотношении	23
7.5 Нахождение общей формулы для a_n	24
7.6 Числа Каталана	24
8 Лекция 20	26
8.1 Комбинаторные игры(определения и примеры)	26
8.2 Теорема о цене игры	27
9 Лекция 21	28
9.1 N и P позиции (выигрышные и проигрышные)	28
9.2 Игра Ним	29
9.3 Задача об угадывании числа	30
9.4 Разрешающие деревья	30
9.5 Примеры задач	31
9.5.1 Задача о взвешивании	31
9.5.2 Задача о связности графа	32
10 Лекция 22	33
10.1 Булевы схемы	33
10.2 Сложение двоичных чисел	34
10.3 Вычисление произвольной функции	35
11 Лекция 23	36
11.1 Функция XOR_n	36
11.2 Глубина схемы	39
11.3 Формулы	41
11.4 Задачи выполнимости	42
12 Лекция 24	43
12.1 Задача выполнимости	43
12.2 Теорема о балансировке булевых функций	44
13 Лекция 24	45
14 Семинар от 23.09.2016	45

1 Лекция 13

(Андрей)

1.1 Вероятность

$\Omega = \{\omega_0, \omega_1, \dots, \omega_n\}$ - пространство элементарных исходов. Будем считать, что это множество конечно.

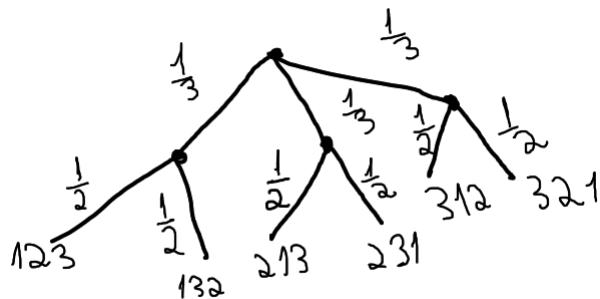
Зададим следующее отображение $P : \Omega \rightarrow [0,1]$, причем $\sum_{i=1}^n P(\omega_i) = 1$. Такое отображение называется вероятностным распределением. Число $P(\omega_i)$ называют вероятностью исхода $\omega_i \in \Omega$.

Событие - это подмножество A пространства элементарных исходов Ω . Вероятность события A : $P(A) = \sum_{x \in A} P(x)$.

Равновероятная модель: $P(w) = \frac{1}{n}$, $P(A) = \frac{|A|}{n}$.

Примеры (в равновероятной модели):

- Найдем вероятность выпадения орла при подбрасывании монеты. Обозначим орла за 1, решку - за 0. Тогда $\Omega = \{0, 1\}$, а $P(1) = P(0) = \frac{1}{2}$.
- Теперь кидаем кубик. Тогда $\Omega = \{0, 1, 2, 3, 4, 5, 6\}$. Найдем вероятность выпадения четного числа. Тогда $A = \{2, 4, 6\}$, и $P(A) = \frac{3}{6}$.
- Теперь подбросили монетку 6 раз. Тогда $\Omega = \{0, 1\}^6$. Пусть событие A - "выпало 3 орла". Тогда $P(A) = \frac{|A|}{|\Omega|} = \frac{C_6^3}{2^6}$.
- Теперь подбросили монету n раз. Тогда $\Omega = \{0, 1\}^n$. Рассмотрим следующее событие A - "в i -ом подбрасывании выпал орел". Честно покажем, почему вероятность этого события равна $\frac{1}{2}$. Так как $A = \{(a_1, \dots, 1, \dots, a_n) : a_i \in [0,1]\}$, то $P(A) = \frac{2^{n-1}}{2^n} = \frac{1}{2}$.
- Случайная перестановка.** Рассмотрим перестановки на трех элементах, то есть $\Omega = S_3$. Мы хотим случайным образом выбрать какую-то перестановку, и так как перестановок всего 6, то каждую перестановку можно задать с вероятностью $\frac{1}{6}$. Но мы поступим по-другому: 1-е число перестановки выберем равновероятно из $\{1, 2, 3\}$, 2-е число выберем равновероятно из оставшихся двух чисел, а 3-е число определится автоматически. Эту светлую мысль хочется записать строго, поэтому давайте изобразим нашу ситуацию в виде дерева.



Здесь на ребрах указана вероятность выбора каждой из цифр. В нашем случае Ω - это листья, а итоговая вероятность выбрать одну перестановку будет равна произведению чисел на ребрах в пути от корня до Ω . В данном случае вероятность каждого исхода равна $\frac{1}{6}$.

1.2 Свойства вероятности

- $P(\emptyset) = 0, P(\Omega) = 1$
- Пусть $A, B \subseteq \Omega, A \cap B = \emptyset$, тогда $P(A \cup B) = P(A) + P(B)$
- $P(\bar{A}) = 1 - P(A)$
- Если $A \subseteq B$, то $P(A) \leq P(B)$

1.3 Оценка объединения

Пусть $A_1, A_2, \dots, A_s \subseteq \Omega$. Тогда $P(A_1 \cup A_2 \cup \dots \cup A_s) \leq P(A_1) + P(A_2) + \dots + P(A_s)$.

Доказательство

По определению $P(\bigcup_{i=1}^n A_i) = \sum_{x \in A_1 \cup \dots \cup A_s} P(x) = [\sum_{x \in A_1} P(x_1)] + [\sum_{x \in A_2 \setminus A_1} P(x_2)] + [\sum_{x \in A_3 \setminus (A_1 \cup A_2)} P(x_3)] + \dots + [\sum_{x \in A_s \setminus (A_1 \cup \dots \cup A_{s-1})} P(x_s)] \leq P(A_1) + P(A_2) + \dots + P(A_s)$.

1.4 Нижняя оценка на числа Рамсея

Докажем следующую теорему: $R(k, k) > 2^{\frac{k-1}{2}}$.

Доказательство

Рассмотрим $n \leq 2^{\frac{k-1}{2}}$, $n \in \mathbb{N}$. Построим граф на n вершинах так, чтобы не было клик и независимых множеств размера k . Возьмем случайный граф. Что это значит? Это значит, что мы рассматриваем следующее вероятностное пространство $\Omega = \{\text{все графы на множестве вершин } 1, 2, \dots, n\}$. Тогда $|\Omega| = 2^{C_n^2}$ (всего ребер в граfe на n вершинах C_n^2 , а различных подмножеств этих ребер как раз $2^{C_n^2}$). Рассмотрим вероятность события A - "в граfe есть клика размера k или независимое множество размера k ".

Рассмотрим произвольное подмножество w множества вершин: $w \subset \{1, 2, \dots, n\}$, $|w| = k$. Тогда рассмотрим следующее событие A_w - "в граfe именно данное фиксированное w является кликой или независимым множеством". Тогда $P(A) = P(\bigcup_{w \subset \{1, \dots, n\}} A_w)$, причем для таких w выполняется, что $|w| = k$. Воспользуемся оценкой объединения (см. пункт 1.3)

и получим: $P(\bigcup_{w \subset \{1, \dots, n\}} A_w) \leq \sum_{w \subset \{1, \dots, n\}} P(A_w)$, причем $P(A_w) = 2 \cdot \frac{2^{C_n^2 - C_k^2}}{2^{C_n^2}} = 2^{1 - C_k^2}$ ($C_n^2 - C_k^2$ - количество ребер в исходном

графе без ребер в w ; умножаем на 2, потому что w может быть как кликой, так и независимым множеством). Тогда $\sum_{w \subset \{1, \dots, n\}} P(A_w) = C_n^k \cdot 2^{1 - C_k^2} = \frac{n \cdot (n-1) \cdot \dots \cdot (n-k+1)}{k!} \cdot 2^{1 - C_k^2} \leq \frac{n^k}{k!} \cdot 2^{1 - C_k^2} \leq \frac{2^{\frac{k \cdot (k-1)}{2}} \cdot 2^{1 - C_k^2}}{k!} = \frac{2}{k!} \leq \frac{1}{3}$. Эта оценка показывает,

что вероятность того, что в граfe нет ни клики размера k , ни независимого множества размера k хотя бы $\frac{2}{3}$. Но это как раз означает, что существует граf на n вершинах, в котором нет ни клики размера k , ни независимого множества размера k , поэтому $R(k, k) > 2^{\frac{k-1}{2}}$, что и требовалось доказать. (скорее всего, Вы офигели с этого доказательства, поэтому попробуйте это перечитать под запись от лектора).

1.5 Формула включений-исключений в вероятностном смысле

Теорема: $P(A_1 \cup A_2 \cup \dots \cup A_n) = P(A_1) + P(A_2) + \dots + P(A_n) - P(A_1 \cap A_2) - \dots - P(A_{n-1} \cap A_n) + \dots (-1)^{n-1} \cdot P(A_1 \cap A_2 \cap \dots \cap A_n)$.

Доказательство

Для начала введем характеристическую функцию $\chi_A : \Omega \rightarrow \{0, 1\}$. Тогда для множества A такого, что $A \subseteq \Omega$, определим:

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Отметим следующие свойства этой функции:

$$\chi_{\bar{A}}(x) = 1 - \chi_A(x)$$

$$\chi_{A \cap B}(x) = \chi_A(x)\chi_B(x)$$

Положим для начала $A = A_1 \cup A_2 \cup \dots \cup A_n$. Применим сначала те же рассуждения, которые мы применяли при доказательстве этой формулы ранее для мощностей множеств.

$$\begin{aligned} \chi_{A_1 \cup A_2 \cup \dots \cup A_n} &= \chi_{\overline{\overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}}} = 1 - \chi_{\overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}} = \\ &= 1 - \chi_{\overline{A_1}}\chi_{\overline{A_2}}\dots\chi_{\overline{A_n}} = 1 - (1 - \chi_{A_1})(1 - \chi_{A_2})\dots(1 - \chi_{A_n}) = \sum_{k=1}^n \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} (-1)^{k+1} \chi_{A_{i_1}}\chi_{A_{i_2}}\dots\chi_{A_{i_k}} \end{aligned}$$

Далее, перейдя к вероятности, имеем: $P(A) = \sum_{x \in A} P(x) = \sum_{x \in \Omega} P(x) \cdot \chi_A(x)$. Теперь, подставляя вместо A объединение множеств $A_1 \cup A_2 \cup \dots \cup A_n$ и вместо $\chi_A(x)$ выражение $1 - (1 - \chi_{A_1})(1 - \chi_{A_2})\dots(1 - \chi_{A_n})$, раскроем скобки и получим

следующую красоту:

$$P(A_1 \cup A_2 \cup \dots \cup A_n) = \sum_{k=1}^n \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} (-1)^{k+1} P(A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_k})$$

2 Лекция 14

(Илья)

2.1 Задача о беспорядках.

Беспорядок - это перестановка $\sigma = (\begin{smallmatrix} 1 & 2 & \dots & n \\ \sigma_1 & \sigma_2 & \dots & \sigma_n \end{smallmatrix}) \forall i \sigma_i \neq i$

Обозначим за $\Omega_n = S_n$ - множество всех перестановок на числах $1, 2, \dots, n$

В равновероятной модели вероятность выбрать случайную перестановку $P(\sigma) = \frac{1}{n!}$, воспользуемся этим, чтобы решить задачу о беспорядках (то есть будем полагать, что все перестановки равновероятны).

$A_n \subseteq \Omega_n$ - это множество беспорядков на n элементах.

Так вот, сама формулировка задачи:

Посчитать вероятность того, что случайно выбранная перестановка окажется беспорядком.

1. Будем считать её через вероятность дополнения. $P(A_n) = 1 - P(\overline{A_n})$

То есть $P(\overline{A_n})$ - это вероятность того, что случайно выбранная перестановка оставила на месте **хотя бы** один элемент от 1 до n

Вероятность "**хотя бы**" очень удобно считать по формуле включений и исключений, введём пару обозначений и так и сделаем.

Обозначим за $B_i \subseteq \Omega_n$ перестановку, которая оставляет элемент i на своём месте.

2. $\overline{A_n} = B_1 \cup B_2 \cup \dots \cup B_n$. Тогда

$$P(\overline{A_n}) = \sum_{k=1}^n (-1)^{k+1} \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} P(B_{i_1} \cap B_{i_2} \cap \dots \cap B_{i_k})$$

Чему равна вероятность $P(B_{i_1} \cap B_{i_2} \cap \dots \cap B_{i_k})$? То есть это вероятность того, что на i_1, i_2, \dots, i_k местах ничего не переставилось, а на остальных переставилось или нет, мы не знаем. Легко понять, что это $\frac{(n-k)!}{n!}$

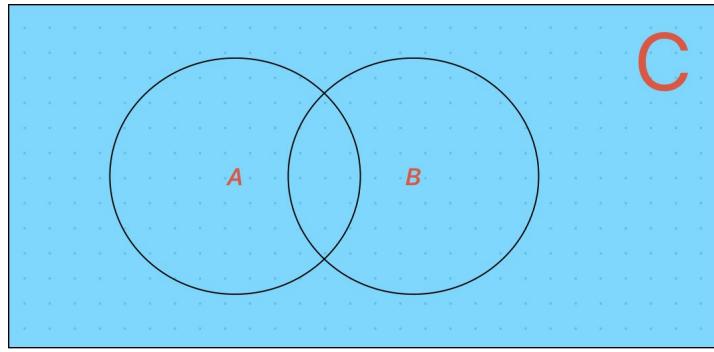
3. Как теперь зная это можно посчитать искомую вероятность дополнения? Очень просто! Надо понять, что второе суммирование производится по всем неподвижным точкам, коих k штук. То есть тогда можно записать так:

$$P(\overline{A_n}) = \sum_{k=1}^n (-1)^{k+1} \cdot \frac{(n-k)!}{n!} \cdot \binom{n}{k} = \sum_{k=1}^n (-1)^{k+1} \cdot \frac{(n-k)!}{n!} \cdot \frac{n!}{k!(n-k)!} = \sum_{k=1}^n (-1)^{k+1} \cdot \frac{1}{k!}$$

Окончательно получаем, что $P(A_n) = 1 - \sum_{k=1}^n (-1)^{k+1} \cdot \frac{1}{k!}$

Вам этот ряд ничего не напоминает? Конечно же это похоже на e^{-1} (А если не похоже, то напомню факт $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$). Тогда $\lim_{n \rightarrow \infty} P(A_n) = \frac{1}{e}$

2.2 Условная вероятность



C - это вероятностное пространство.

Пусть $A \subseteq C$ - это событие, которое произошло ($P(A) > 0$), $B \subseteq C$ - это событие, которое мы хотим посчитать, при условии того, что событие A уже произошло.

Формула условной вероятности:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Пример:

Задача: У нас имеется 9 коробочек(они лежат на столе пронумерованные). Разрешается делать пару вещей:

1. Выбрать равновероятно одну коробку.
2. Равновероятно положить или не положить в неё монетку.

Известно после открытия, что в первых 8 коробках монетки нет. **Какова вероятность того, что в 9 коробке лежит монетка?**

Решение:

1. Пусть A - событие 'монет нет в первых 8 коробках', а B - событие 'монета в 9-ой коробке', тогда мы хотим найти $P(B|A) = \frac{P(A \cap B)}{P(A)}$
2. В любой задаче на вероятность надо сначала выбрать вероятностное пространство. Пусть $C = \{(i,j) | 1 \leq i \leq 9, j = 0,1\}$, где i - номер коробки, j - есть монета или нет.
Тогда понятно, что $P((i,j)) = \frac{1}{18}$ (это можно легко осознать, нарисовав дерево вероятностное)
3. Теперь мы имеем, что $A = \{(i,0) | i = 1,2,\dots,8\} \cup \{(9,1)\}$, $B = \{(9,1)\}$. Значит $|A| = 10$ и $|B| = 1$. Ну тогда искомая вероятность $P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{1}{10}$

Перепишем формулу немного по-другому: $P(B|A) \cdot P(A) = P(A \cap B)$. Эту формулу можно обобщить на произвольное количество событий.

Теорема умножения.

Пусть имеется $A_1, A_2, \dots, A_n \subseteq C$ - события положительной вероятности. Тогда $P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_n) \cdot P(A_{n-1}|A_n) \cdot P(A_{n-2}|A_{n-1} \cap A_n) \cdot \dots \cdot P(A_1| \bigcap_{i=2}^n A_i)$ Более строго это утверждение можно доказать по индукции, но оставим это в качестве упражнения.

2.3 Формулы Байеса и полной вероятности.

Формула Байеса. Пусть $A, B \subseteq C$ - события положительной вероятности, тогда $P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$

Доказательство этого факта очевидно.

Пример про болезни.

Предположим есть болезнь и мы хотим знать болен ли человек или нет. Существует 2 теста: дешёвый(менее точный) и дорогой(более точный). Мы можем отправить человека сначала на дешёвое тестирование. Если тест показал положительный результат, то отправляем его на дорогое тестирование, которое почти гарантированно может дать точный результат, иначе отправляем его сразу домой. **Хотим понять с какой вероятностью мы отпустим больного человека домой.**

Решение:

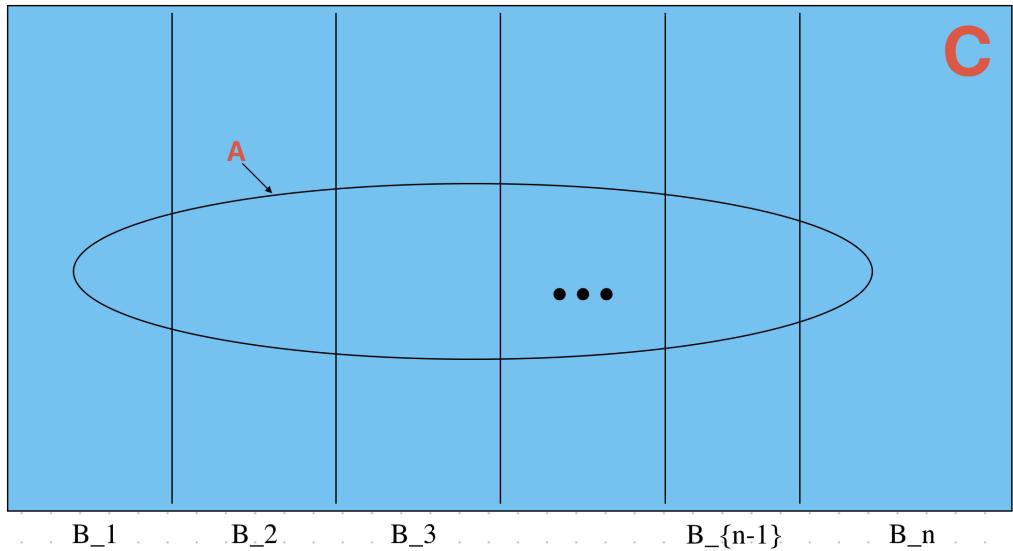
1. Пусть B - событие 'человек болен', A - событие 'дешёвый тест показал "не болен"', тогда нужно посчитать $P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$. Давайте посчитаем искомую вероятность через две другие: $P(\bar{A}|B)$ и $P(B|\bar{A})$.
2. Заметим, что $P(B|\bar{A})$ мы знаем, потому что это в точности то, что нам выдаёт дорогой тест, но как же посчитать $P(\bar{A}|B)$? По формуле Байеса!
- Окей. Осталось понять, как считать искомую вероятность, зная $P(\bar{A}|B)$.
- Заметим, что $P(A|B) + P(\bar{A}|B) = \frac{P(\bar{A} \cap B) + P(A \cap B)}{P(B)} = 1$. Это легко увидеть, так как сумма вероятностей несовместных событий есть вероятность объединения этих событий. Получаем в числителе $P(B)$.
- Ну и всё, зная $P(A|B)$, находим искомую вероятность по той же формуле Байеса.

Формула полной вероятности. Пусть $B_1, B_2, \dots, B_n \subseteq C$ - это события с положительной вероятностью и $B_1 \sqcup B_2 \sqcup \dots \sqcup B_n = C$ ($B_i \cap B_j = \emptyset, i \neq j$)

Тогда вероятность произвольного события $A \subseteq C$ равна $\sum_{i=1}^n P(A|B_i) \cdot P(B_i)$.

Доказательство:

1. Сделаем наглядный рисунок, демонстрирующий разбиение вероятностного пространства C на не пересекающиеся множества.



2. Тогда $A = (A \cap B_1) \sqcup (A \cap B_2) \sqcup \dots \sqcup (A \cap B_n)$, откуда $P(A) = \sum_{i=1}^n (A \cap B_i) = \sum_{i=1}^n P(A|B_i) \cdot P(B_i)$.

Ч.Т.Д.

Пример о распределении на рёбрах графа. Предположим у нас есть граф $G = (V, E)$, $|V| = n$, $\forall v \in V, \deg(v) = d$ (G - регулярный граф, потому что степень каждой вершины константа.)

Мы хотим случайно выбрать ребро этого графа. Есть несколько способов это сделать:

1. Равновероятно выбрать из множества рёбер.
2. Сначала равновероятно выбрать вершину этого графа, а потом равновероятно выбрать ребро, смежное этой вершине.

Формально это разные конструкции, но с точки зрения распределения на рёбрах это эквивалентные построения, давайте покажем это.

В случае 1:

$P(e) = \frac{1}{\frac{n \cdot d}{2}} = \frac{2}{nd}$ (по лемме о рукопожатиях, так как сумма степеней вершин есть удвоенное число рёбер)

В случае 2:

Пусть B_i - событие 'была выбрана i -ая вершина', тогда $P(e) = \sum_{i=1}^n P(e|B_i) \cdot P(B_i) = \sum_{i=1}^n P(e|B_i) \cdot \frac{1}{n}$

Тут надо подумать, когда $P(e|B_i)$ не ноль. Ну она не ноль только при двух конкретных i . Поэтому $P(e) = \frac{1}{d} \cdot \frac{1}{n} + \frac{1}{d} \cdot \frac{1}{n} = \frac{2}{dn}$

Далее идёт ещё один пример с болезнью, но он идентичен первому, просто с числами, не вижу смысла вставлять его ещё раз.

2.4 Независимые события.

Сначала дадим парочку определений, которые нужно отличать.

Несовместные события A и $B \subseteq C$ называются таковыми, если $P(A \cap B) = 0$.

Независимые события A и $B \subseteq C$ называются таковыми, если $P(A \cap B) = P(A) \cdot P(B)$.

Замечание: Если $P(B) > 0$, то независимость A и B равносильна $P(A|B) = \frac{P(A \cap B)}{P(B)} = P(A)$.

Далее шёл пример с подбрасыванием монетки, но он очень простой.

Что если я хочу определить независимость большего числа событий? Пусть $A_1, A_2, \dots, A_n \subseteq C$ называются независимыми в совокупности, если $\forall i_1 < i_2 < \dots < i_k P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_{i_1}) \cdot P(A_{i_2}) \cdot \dots \cdot P(A_{i_k})$

Далее опять пример с монеткой, но он опять же довольно очевидный.

3 Лекция 15

(Никита)

3.1 Раскраски графов. Хроматическое число графа.

Для начала введём несколько обозначений, которые будут использоваться на протяжении всей лекции:

- $G = (V, E)$ - неориентированный граф без петель и кратных рёбер.
- n - количество рёбер в графе G , $|V|$.
- $\deg v$ - степень вершины v в G .
- $\Delta(G)$ - максимальная степень в G .
- $\omega(G)$ - размер максимальной клики в G (кликовое число).
- $\alpha(G)$ - число независимости в G , размер максимального независимого множества в G .

Дадим определение раскраски:

Раскраска (правильная) графа G в k цветов - это

$$f : V \rightarrow \{1, 2, \dots, k\} : \forall v_1, v_2 \in V : \{v_1, v_2\} \in E \Rightarrow f(v_1) \neq f(v_2)$$

Хроматическое число G - это минимальное число цветов, в которые можно раскрасить G .

Обозначение: $\chi(G)$.

Замечание. $\chi(G) = \min\{k \in \mathbb{N} \mid V = V_1 \sqcup V_2 \sqcup \dots \sqcup V_k, \forall V_i \text{ - независимое множество}\}$.

Если мы разбили граф на независимые множества - красим каждое независимое множество в свой цвет, и наоборот - если смогли раскрасить граф в k цветов, то все вершины одного цвета обязаны образовывать независимое множество. Тогда понятно, что хроматическое число - это и есть минимальное кол-во множеств в разбиении вершин на независимые множества.

Свойства хроматического числа:

1. $\chi(G) \geq \omega(G)$. На клику требуется $\omega(G)$ цветов, т.к. каждые две вершины в ней соединены ребром.
2. $\chi(G) \geq \frac{n}{\alpha(G)}$.

Доказательство:

$\chi(G) = k$ и $V = V_1 \sqcup V_2 \sqcup \dots \sqcup V_k$, где V_i - независимое множество.

$$|V_i| \leq \alpha(G) \Rightarrow n \leq k \cdot \alpha(G) \Rightarrow k \geq \frac{n}{\alpha(G)} \blacksquare$$

3. $\chi(G) \leq n - \alpha(G) + 1$.

Доказательство:

$\chi(G) = k$ и $V = V_1 \sqcup V_2 \sqcup \dots \sqcup V_k$, где V_i - независимое множество.

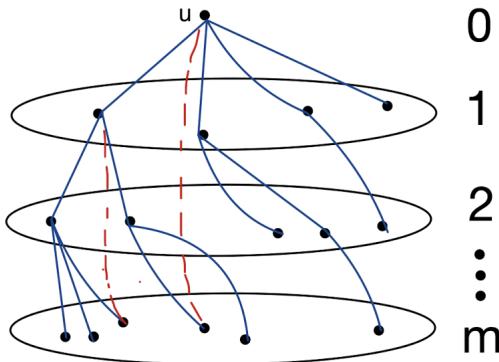
Красим максимальное независимое множество в какой-то цвет, а каждую из оставшихся $n - \alpha(G)$ вершин в свой уникальный цвет. Тогда $\chi(G) \leq n - \alpha(G) + 1$ ■

4. $\chi(G) \leq \Delta(G) + 1$

Доказательство:

Не умоляя общности, G - связен (иначе можно провести описанное ниже рассуждение для каждой компоненты связности и выбрать в качестве хроматического числа максимальное из хроматических чисел компонент).

Тогда выделим в G остовное дерево (напомним, что это всегда можно сделать для связного графа). Подвесим его за произвольную вершину u (значит выберем её в качестве корня).



Теперь будем действовать по такому алгоритму: начиная с последнего слоя будем красить вершины в цвета $\{1, 2, \dots, \Delta(G)\}$. Для каждой вершины x будем смотреть, с какими уже покрашенными она соединена (учитывая и рёбра вне дерева - обозначены красным на картинке). Если x - не корень, то есть хотя бы одно ребро в верхний слой, то есть рёбер в слоях с уже покрашенными вершинами строго меньше, чем $\Delta(G)$. Значит вершину x можем покрасить в какой-то из оставшихся цветов (один точно найдётся).

Теперь рассмотрим ситуацию с корнем. Может получиться так, что корень соединён с $\Delta(G)$ вершинами из нижнего покрашенного слоя (ему ведь не нужно быть соединённым ребром с кем-то выше), так что в нашей оценке мы добавляем 1, чтобы в таком случае корень можно было покрасить в $(\Delta(G) + 1)$ -й цвет ■

Следствие:

Если G связен и $\exists w \in V : \deg w < \Delta(G)$, то $\chi(G) \leq \Delta(G)$.

Доказательство:

Повторим описанное выше рассуждение, только подвесим остовное дерево именно за w . Тогда в любом случае получится так, что на последнем шаге раскраски рёбер в покрашенные вершины строго меньше чем $\Delta(G)$ (просто по условию). Аналогично второму абзацу предыдущего доказательства может понадобиться ещё один цвет. То есть $\chi(G) \leq \Delta(G)$ ■

Примеры:

(a) $G = K_n$ (полный граф). Тогда $\chi(G) = n$, $\Delta(G) = n - 1$.

(b) $G = C_{2s+1}$ (нечётный цикл). Тогда $\chi(G) = 3$, $\Delta(G) = 2$.

Теорема Брукса (без доказательства)

Если G - связен и $\chi(G) = \Delta(G) + 1$, то G "изоморфен" K_n или C_{2s+1} .

Определение изоморфизма графов

Графы $G = (V_1, E_1)$ и $G = (V_2, E_2)$ изоморфны (обозначается как $G_1 \cong G_2$), если $\exists f : V_1 \rightarrow V_2$ такая, что:

- f - биекция
- $\forall u, v \in V_1 : \{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$

3.2 Конструкция Зыкова-Мыцельского графа без треугольников со сколь угодно большим хроматическим числом.

Теорема Зыкова-Мыцельского

Пусть $k \in \mathbb{N}$. Тогда существует граф G : $\omega(G) = 2$, $\chi(G) = k$.

Доказательство:

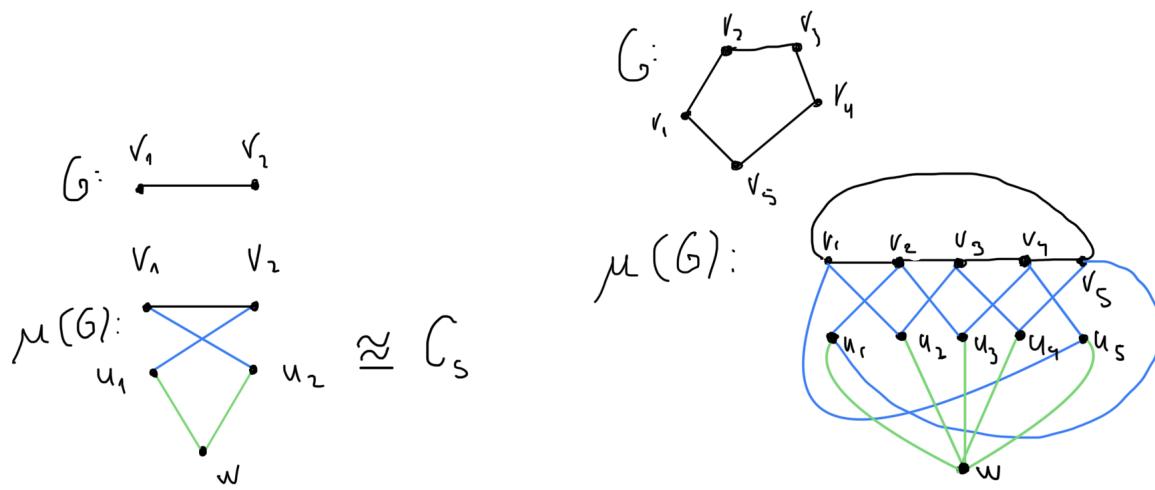
Мыцельскийан

Граф $G \rightarrow \mu(G)$, где $G = (\{v_1, v_2, \dots, v_n\}, E)$

Вершины в $\mu(G)$: $v_1, v_2, \dots, v_n, u_1, u_2, \dots, u_n, w$ - всего $2n + 1$ штук.

Рёбра в $\mu(G)$: если $\{v_i, v_j\} \in E(G)$, то $\{v_i, v_j\}, \{u_i, v_j\}, \{v_i, u_j\} \in E(\mu(G))$. А также $\{w, u_i\} \in E(\mu(G)) \forall i$.

Примеры:



Утверждается, что рассматривая последовательность графов $G_1 = K_2$, $G_2 = \mu(G_1)$, $G_3 = \mu(G_2) = \mu(\mu(G_1))$, ..., $G_s = \mu(G_{s-1})$, ..., получим, что G_s не содержит треугольников и $\chi(G_s) = s + 1$.

1. Докажем, что G_s не содержит треугольников:

Пусть s - минимальное такое число (≥ 3), что в $G_s = \mu(G_{s-1})$ есть треугольник. По построению могут образовываться лишь вершины v_i, v_j, u_l (именно такой набор, потому что две вершины u не могут быть в треугольнике, так как не соединены ребром; w также не может - иначе две вершины u должны быть соединены ребром) причём $l \neq i, j$ по построению.

Раз в $G_s = \mu(G_{s-1})$ есть ребро $\{u_l, v_i\}$, то в исходном графе G_{s-1} было ребро $\{v_i, v_l\}$. Аналогично, если в $G_s = \mu(G_{s-1})$ есть ребро $\{u_l, v_j\}$, то в исходном графе G_{s-1} было ребро $\{v_j, v_l\}$. Но в G_{s-1} есть и ребро $\{v_i, v_j\}$, значит рёбра $\{v_i, v_l\}, \{v_j, v_l\}, \{v_i, v_j\}$ образуют треугольник в графе G_{s-1} . Противоречие, потому что s оказалось не минимальным таким числом.

2. Докажем, что $\chi(G_s) = s + 1$ индукцией по s :

База проверена выше.

Переход $s - 1 \rightarrow s$.

Известно, что $\chi(G_{s-1}) = s$, нужно доказать, что $\chi(G_s) = \chi(\mu(G_{s-1})) = s + 1$

Можно считать, что $s \geq 3$. Тогда покрасим u_i в $(s + 1)$ -й цвет, w в 1-й цвет, а v_i уже покрашены в s первых цветов. Покажем что меньшим числом мы не обойдёмся.

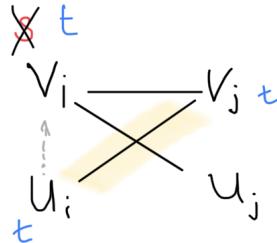
Нам известно, что $\chi(\mu(G_{s-1})) \geq s$, т.к. G_{s-1} содержится в $\mu(G_{s-1})$ как подграф. Покажем, что ровно s цветами мы не обойдёмся.

От противного: пусть $\chi(\mu(G_{s-1})) = s$. Будем считать, что w покрашена в s -ый цвет. Тогда среди u_i нет цвета s (потому что w соединена со всеми u_i ребром). Но по предположению индукции цвет s должен найтись среди вершин v_j (их нельзя было покрасить меньшим чем s кол-вом цветов). Тогда есть некоторое количество вершин v_j цвета s . Перекрасим их в цвета по такому правилу: v_j красится в цвет, который имеет вершина u_j (для наглядности можно смотреть на пример 2 выше). Утверждается, что у нас получилась правильная раскраска вершин подграфа $G_{s-1}(v_1, \dots, v_n)$ в $s - 1$ цвет.

Замечание. Несмотря на популярное заблуждение, раскраска в k цветов всё ещё будет называться таковой, даже если мы не использовали все цвета. То есть раскраской в 4 цвета может называться и двухцветная раскраска (два цвета могут просто не использоваться).

Продолжим. Докажем, что получилась именно правильная раскраска вершин подграфа $G_{s-1}(v_1, \dots, v_n)$ в $s-1$ цвет.

От противного: пусть после этого перекрашивания пусть соединены ребром какие-то одноцветные v_i и v_j (пусть их цвета - t). То есть до перекрашивания ровно одна из них была цвета s (понятно, что не обе, т.к. это была бы неправильная раскраска всего графа). Ну v_i была цвета s , а v_j была покрашена в какой-то цвет t . Мы не знаем, какого цвета в Мыцельскиане была u_j , но точно знаем, что u_i была цвета t , потому что мы в него перекрасили v_i по нашему правилу. Противоречие, т.к. в Мыцельскиане есть одноцветное ребро $\{u_i, v_j\}$. То есть $\chi(G_s) > s \Rightarrow \chi(G_s) \geq s+1$, а как раскрасить в $s+1$ цвет мы показали выше. Переход доказан, значит $\chi(\mu(G_{s-1})) = s+1$ ■



Чем интересен граф Зыкова-Мыцельского? Смотрите: если мы знаем, что у графа бесконечно большое кликовое число, то можем сказать, что и хроматическое число графа бесконечно большое (оно как минимум равно кликовому). Однако обратное неверно. Мы построили граф, в котором при бесконечно большом хроматическом числе кликовое число равно 2.

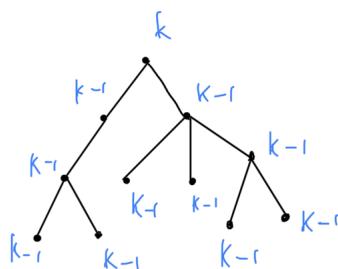
3.3 Хроматический многочлен.

Хроматический многочлен. Функция от числа цветов, возвращающая количество способов раскрасить (естественно правильно) фиксированный граф G в k цветов.

Обозначение: $\chi_G(k)$.

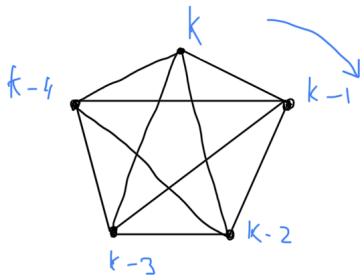
Примеры:

- G - дерево. $\chi_G(k) = k \cdot (k-1)^{n-1}$



Для корня цвет выбираем k способами, а для любой другой вершины есть запрет одного цвета от вершины из над-слоя. То есть $k-1$ способов для такой вершины. Отсюда такой вид многочлена.

- G - полный граф. $\chi_G(k) = k \cdot (k-1) \cdot (k-2) \cdot \dots \cdot (k-n+1)$



Выбираем для какой-то вершины цвет k способами, потому выбираем любую другую. Для неё уже $k - 1$ способ выбрать цвет, т.к. она соединена с покрашенной (в полном графе любые две соединены). И так далее.

- G - независимое множество. $\chi_G(k) = k^n$. Очевидно, т.к. никакие две вершины не соединены ребром.

Определим две операции:

- удалить $\{u, v\}$. Обозначение: $G - uv$.
- склеить u и v . Обозначение: $G \cdot uv$. Что это значит: мы объединяем вершины u и v в одну, и все ребра, которые вели в u и v теперь будут вести в эту новую вершину (если получилось кратное ребро - заменим его на одно).

Утверждение. $\chi_{G-uv}(k) = \chi_G(k) + \chi_{G\cdot uv}(k)$, если $\{u, v\} \in E(G)$.

Доказательство:

Удалим $\{u, v\}$. Теперь посчитаем кол-во способ раскрасить $G - uv$:

1. u и v раскрашены в разные цвета. Тогда нам ничего не мешает вернуть ребро $\{u, v\}$ и раскраска останется правильной. Тогда очевидна биекция: таких раскрасок ровно столько же, сколько у графа G .
2. u и v раскрашены в один цвет. Тогда рассмотрим граф со склеенными вершинами u и v в одну. Тогда любая его раскраска соответствует правильной раскраске исходного графа, где u и v раскрашены в один цвет (точнее почти правильной, "неправильным" будет только ребро $\{u, v\}$). И наоборот. Вспоминаем, что ребра $\{u, v\}$ уже нет, тогда устанавливаем биекцию: таких раскрасок ровно столько же, сколько у графа $G \cdot uv$.

Итого: $\chi_{G-uv}(k) = \chi_G(k) + \chi_{G\cdot uv}(k)$ ■

Теорема Уитни о хроматическом многочлене и его свойствах

$\chi_G(k)$ - многочлен вида $k^n - a_1 k^{n-1} + a_2 k^{n-2} - \dots + (-1)^{n-r} a_{n-r} k^r$, где $a_i \in \mathbb{N}$, $a_1 = |E(G)|$, r - количество компонент связности.

Доказательство:

Полная индукция по числу ребер в графе.

База очевидна: если 0 ребер, то граф - независимое множество, $\chi_G(k) = k^n$ (см. пример выше).

Переход.

$\chi_G(k) = \chi_{G-uv}(k) - \chi_{G\cdot uv}(k)$, если $\{u, v\} \in E(G)$ по доказанному выше утверждению. Кол-во ребер в графах $G - uv$ и $G \cdot uv$ меньше чем в G , поэтому по индукционному предположению для их хроматических многочленов верны все свойства.

Будем считать, что в $|E(G)| = m$

В графе $G - uv$ при удалении одного ребра количество компонент связности может или не измениться, или увеличиться на 1.

В графе $G \cdot uv$ при склеивании двух вершин количество компонент связности не меняется.

$\chi_{G-uv}(k)$ выглядит как $k^n - (m-1)k^{n-1} + \dots + (-1)^{n-r-1} b_{n-r-1} k^{r+1} + (-1)^{n-r} b_{n-r} k^r$. При этом имеем в виду, что b_{n-r} -й коэффициент может быть равен 0 в случае увеличения количества компонент на 1. Будем считать 0 как положительным, так и отрицательным.

$\chi_{G\cdot uv}(k)$ выглядит как $k^{n-1} - c_1 k^{n-2} + \dots + (-1)^{n-r-1} c_{n-r-1} k^r$

При вычитании многочлена с натуральными a_i из многочлена с натуральными b_i очевидно, что получится также многочлен с натуральными c_i . Старший коэф-т при k^n действительно будет 1. Коэф-т при k^{n-1} будет равен $m-1+1 = m$. Теперь проверим, что будет знакочередование: в многочлене $\chi_{G-uv}(k)$ коэф-ты знакочередуются и можно заметить, что в вычитаемом многочлене $\chi_{G\cdot uv}(k)$ при симметричных степенях знаки противоположны. Поэтому при вычитании из отрицательного коэф-та положительного он будет оставаться отрицательным, а при вычитании из положительного коэф-та отрицательного он будет оставаться положительным ■

Как с помощью хроматического многочлена найти хроматическое число? Нужно просто подставлять вместо k натуральные числа. Получим:

$\chi_G(1) = \chi_G(2) = \dots = \chi_G(\chi(G) - 1) = 0$ - то есть до какого-то натурального будет получаться значение 0, а первое натуральное m при котором многочлен равен ненулевому числу $\chi_G(m)$ и будет $\chi(G)$.

Упражнение. Если G - связен, то $\chi_G(0) = 0$, причём кратность нуля = 1.

4 Лекция 16

(Андрей)

4.1 Математическое ожидание

Пусть имеем $\Omega = \{\omega_1, \dots, \omega_k\}$ - вероятностное пространство с вероятностным распределением p_1, \dots, p_k соответственно, причем понятно, что $\sum_{i=1}^k p_i = 1$. Далее, назовем *случайной величиной* функцию $f : \Omega \rightarrow \mathbb{R}$ и на $\omega_1, \omega_2, \dots, \omega_k$ f принимает значения $\{a_1, a_2, \dots, a_k\}$.

Предположим, что выбор случайного элемента из Ω повторяется n раз. Если n достаточно большое, то случайная величина f примет значение a_1 примерно $p_1 \cdot n$ раз, значение a_2 — примерно $p_2 \cdot n$ раз, и так далее, значение a_k — примерно $p_k \cdot n$ раз. Подсчитаем теперь примерное среднее арифметическое значений случайной величины f в этих экспериментах:

$$\frac{a_1 p_1 n + a_2 p_2 n + \dots + a_k p_k n}{n} = a_1 p_1 + a_2 p_2 + \dots + a_k p_k$$

Такую сумму $\sum_{i=1}^k f(\omega_i)p_i$ называют *математическим ожиданием* случайной величины f и обозначают $E[f]$.

Пример:

Давайте покидаем кубик что ли. Имеем $\Omega = \{1, 2, 3, 4, 5, 6\}$ (все исходы равновероятны). Положим f - число очков, которое выпадает при одном броске кубика. Тогда $E[f] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} = \frac{21}{6} = 3,5$.

Докажем лемму о линейности мат. ожидания.

Лемма: $E[f + g] = E[f] + E[g]$ (f, g - случайные величины).

Доказательство: $E[f + g] = \sum_{i=1}^k (f + g)(\omega_i)p_i = \sum_{i=1}^k f(\omega_i)p_i + \sum_{i=1}^k g(\omega_i)p_i = E[f] + E[g]$.

Пример: задача о днях рождения.

Дано: $n = 28$ - количество людей, случайная величина f - число пар людей, у которых день рождения в один и тот же день.

Утверждается: $E[f] > 1$.

Доказательство:

Для начала поймем, что $\Omega = \{1, 2, \dots, 365\}^n$ (распределение равновероятно). Теперь делаем финт ушами и вводим следующую функцию:

$$g_{ij} = \begin{cases} 1, & \text{если } i\text{-ый человек и } j\text{-ый человек родились в один день} \\ 0, & \text{иначе} \end{cases}$$

Тогда $f = \sum_{1 \leq i < j \leq 28} g_{ij}$. Тогда $E[f] = \sum_{1 \leq i < j \leq 28} E[g_{ij}]$. Заметим, что $E[g_{ij}] = 1 \cdot P$, где P - вероятность того, что два человека с номерами i и j , родились в один день. Она равна $\frac{365^{n-1}}{365^n} = \frac{1}{365}$. И, наконец, $E[f] = \frac{n(n-1)}{2} \cdot \frac{1}{365}$ (для тех, кто не понял, откуда взялся множитель с n , то кол-во пар равно $\frac{n(n-1)}{2}$). Так как $n = 28$, то $E[f] = \frac{378}{365} > 1$.

4.2 Неравенство Маркова

Теорема (неравенство Маркова): Пусть $f : \Omega \rightarrow \mathbb{R}_{\geq 0}$ - неотрицательная случайная величина и пусть некоторое число $\alpha > 0$. Тогда $P(f \geq \alpha) \leq \frac{E[f]}{\alpha}$.

Доказательство:

Жестко идем по определению: $E[f] = p_1 f(\omega_1) + p_2 f(\omega_2) + \dots + p_k f(\omega_k)$. Дальше хотим сделать следующее:

$$f(\omega_i) = \begin{cases} 0, & \text{если } f(\omega_i) < \alpha \\ \alpha, & \text{если } f(\omega_i) \geq \alpha \end{cases}$$

При такой замене для исходного выражения выполнится следующее неравенство:

$$E[f] = p_1 f(\omega_1) + p_2 f(\omega_2) + \dots + p_k f(\omega_k) \geq \alpha(p_{i1} + p_{i2} + \dots + p_{ij}) = \alpha P(f \geq \alpha)$$

Ч.Т.Д.

Пример: вероятностный алгоритм.

Пусть имеем некоторый алгоритм A_1 , который обладает следующими свойствами:

- Он всегда корректно работает.
- Работает **в среднем** за время $T = O(n^2)$.
- Иногда сильно тормозит.

Хотим: Модифицировать его так, чтобы новый алгоритм A_2 работал следующим образом:

- Он может ошибиться в 0,01% случаев.
- Работает всегда за $O(n^2)$.

Тогда сделаем следующие действия:

- Запустим A_1 на время $10000 \cdot T$.
- Если получили ответ, то выведем его
- Иначе выведем 0.

Пусть случайная величина f - реальное время работы нашего алгоритма.

Тогда имеем $E[f] = T$ (так как T - среднее время работы). Но в таком случае $P(f \geq 10000T) \leq \frac{E[f]}{10000T} = \frac{T}{10000T} = \frac{1}{10000} = 0,01\%$ случаев. Получили требуемый алгоритм.

4.3 Дисперсия

Дисперсия случайной величины $f : \Omega \rightarrow \mathbb{R}$ - это $D[f] = E([f - E\{f\}]^2)$

Утверждение: $D[f] = E[f^2] - (E[f])^2$

Доказательство:

$$D[f] = E([f - E\{f\}]^2) = E(f^2 - 2fE[f] + (E[f])^2) = E[f^2] + E(-2fE[f]) + E\{(E[f])^2\} = E[f^2] - 2E[f]E[f] + (E[f])^2 = E[f^2] - (E[f])^2$$

Ч.Т.Д.

4.4 Неравенство Чебышева

Пусть f - произвольная случайная величина, число $\alpha > 0$. Тогда $P(|f - E[f]| \geq \alpha) \leq \frac{D[f]}{\alpha^2}$

Доказательство:

Рассмотрим случайную величину $g = (f - E[f])^2$. По неравенству Маркова:

$$P(g \geq \alpha^2) \leq \frac{E[g]}{\alpha^2}$$

Так как $P(g \geq \alpha^2) = P(|f - E[f]| \geq \alpha)$ и $E[g] = D[f]$ (из определения дисперсии), то:

$$P(|f - E[f]| \geq \alpha) \leq \frac{D[f]}{\alpha^2}$$

Ч.Т.Д.

4.5 Общий случай вероятностного метода

Лемма: Пусть $E[f] = c$, при этом имеем Ω и $f : \Omega \rightarrow \mathbb{R}$. Тогда:

- $\exists \omega_{min} \in \Omega : f(\omega_{min}) \leq c$
- $\exists \omega_{max} \in \Omega : f(\omega_{max}) \geq c$

Доказательство:

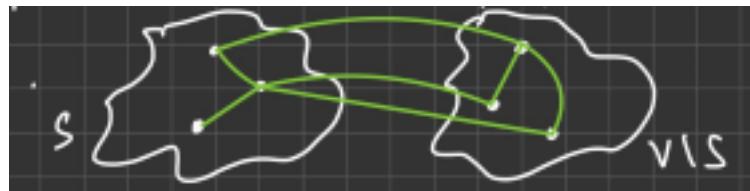
По определению $E[f] = p_1 f(\omega_1) + p_2 f(\omega_2) + \dots + p_k f(\omega_k)$.

Пойдем от противного. Пусть $f(\omega_i) \geq c \forall i$, то $E[f] > c(p_1 + p_2 + \dots + p_k) = c$. Получили противоречие с условием леммы, следовательно, $\exists \omega_{min} \in \Omega : f(\omega_{min}) \leq c$. Для второго случая доказательство аналогично.

Ч.Т.Д.

Пример: Разрезы в графах

Рассмотрим простой неориентированный граф $G = (V, E)$. *Разрезом* графа называется разбиение множества его вершин на два непересекающихся подмножества: $V = V_1 \sqcup V_2$. Мы говорим, что ребро попадает в разрез, если один его конец лежит в V_1 , а другой в V_2 . Размером разреза называется число рёбер, попадающих в разрез.



Утверждение: В любом графе есть разрез размера $\frac{|E|}{2}$.

Доказательство:

Пусть S - случайное подмножество V . Имеем случайный разрез графа $(S, V \setminus S)$. Введем случайную величину $g_e = \begin{cases} 1, & \text{если ребро } e \text{ лежит в разрезе} \\ 0, & \text{иначе} \end{cases}$

Далее определим случайную величину f как $f = \sum_{e \in E} g_e$. Тогда $E[f] = \sum_{e \in E} E[g_e] = \sum_{e \in E} \frac{1}{2} = \frac{|E|}{2}$ ($E[g_e] = \frac{1}{2}$, из определения мат. ожидания).

Ч.Т.Д.

5 Лекция 17

(Илья)

5.1 Вероятностный метод. Разрезы в графах.

Пусть нам дан неориентированный граф $G = (V, E)$ без кратных рёбер и петель и разрез $S \subseteq V$ величиной $|E(S, V \setminus S)|$ (более подробно о том, что такое разрез в графе и его величина можно прочитать в конце 16 лекции)

Теорема: В G существует разрез величиной не меньше, чем $\frac{|E|}{2}$.

Данная теорема была доказана на прошлой лекции, поэтому опять отсылаю вас прочитать её.

Так вот, к чему это я веду, оказывается, что можно немного усилить предыдущую теорему.

Теорема: Пусть $|V| = 2n$ - чётное число вершин графа (аналогично можно доказать и для нечётного числа вершин). Тогда в G существует разрез не меньше, чем $\frac{|E|n}{2n-1}$.

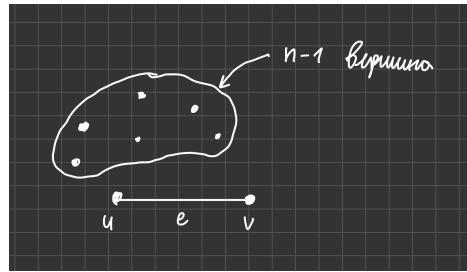
Доказательство:

1. Пусть $\Omega = \{S \subseteq V | |S| = n\}$ - наше вероятностное пространство и все события равновероятны, тогда понятно, что $|\Omega| = \binom{2n}{n}$.

Пусть f - случайная величина и $f : \Omega \rightarrow \mathbb{R}$ и $f(S) = |E(S, V \setminus S)|$. То есть она разрезу ставит в соответствие его величину. Тогда давайте воспользуемся вероятностным методом, как мы доказывали предыдущую теорему, и просто посчитаем $E[f]$.

2. Давайте зададим индикаторную функцию $g_e = \begin{cases} 1, & \text{если ребро } e \in E(S, V/S) \\ 0, & \text{если ребро не лежит в разрезе} \end{cases}$

Тогда $f = \sum_{e \in E} g_e$, а значит, что $E[f] = \sum_{e \in E} P(g_e = 1)$. Посчитаем, чему равна вероятность того, что ребро входит в разрез графа. Чтобы осознать это давайте сделаем поясняющий рисунок.



Вероятность того, что ребро e попадёт в разрез, означает, что мы сможем разделить множество вершин из $2n$ вершин на 2 непересекающихся подмножества вершин по n вершин в каждом, причём вершины соединённые ребром e лежат в разных подмножествах. Давайте посмотрим, сколькими способами мы можем это сделать. Понятно, что я могу посчитать количество способов выбрать одно такое подмножество на n вершинах, тогда второе задаётся однозначно.

Вот пусть у меня есть вершины u и v и они соединены ребром e , тогда зафиксировав вершину u мы получим, что количество способов дополнить её до множества размером n , причём, чтобы v не лежало в нём, равно $\binom{2n-2}{n-1}$ (если до сих пор не понятно, то прочитайте вдумчиво, что я тут написал). Но данное количество способов надо домножить на 2, потому что в самом начале я мог зафиксировать не вершину u , а вершину v . Итого количество подходящих разбиений множества для того, чтобы ребро e попало в разрез равно $2\binom{2n-2}{n-1}$.

Понятно, что всего разбиений множества из $2n$ вершин на 2 подмножества из n вершин это $\binom{2n}{n}$.

$$\text{Итого, } P(g_e = 1) = \frac{2\binom{2n-2}{n-1}}{\binom{2n}{n}} = \frac{2 \cdot (2n-2)! \cdot n! \cdot n!}{(n-1)! \cdot (n-1)! \cdot (2n)!} = \frac{2 \cdot n^2}{2n(2n-1)} = \frac{n}{2n-1}$$

3. Значит, $E[f] = \sum_{e \in E} P(g_e = 1) = \frac{|E|n}{2n-1}$. Тогда по лемме о вероятностном методе (прочтите прошлую лекцию, если не понятно, о чём я) получаем, что существует разрез в графе величиной $\frac{|E|n}{2n-1}$

Ч.Т.Д.

Упражнение: Если $V = 2n + 1$, то существует разрез в графе величиной не меньше, чем $\frac{|E|(n+1)}{2n+1}$

5.2 Независимые случайные величины

Пусть $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ - вероятностное пространство.

Определение: Случайные величины $X, Y : \Omega \rightarrow \mathbb{R}$ называются независимыми, если $\forall x, y \in \mathbb{R}$ события $\{\omega \in \Omega | X(\omega) = x\}$ и $\{\omega \in \Omega | Y(\omega) = y\}$ независимы. (напомню, что события A, B называются независимыми, если $P(A \cap B) = P(A) \cdot P(B)$).

Замечание: X_1, X_2, \dots, X_n независимы, если $\forall x_1, x_2, \dots, x_n \in \mathbb{R} P((X_{i_1} = x_{i_1}) \cap (X_{i_2} = x_{i_2}) \cap \dots \cap (X_{i_n} = x_{i_n})) = P(X_{i_1} = x_{i_1}) \cdot P((X_{i_2} = x_{i_2}) \cdot \dots \cdot P((X_{i_n} = x_{i_n}))$

Теорема: Если случайные величины X, Y независимы, то $E[X \cdot Y] = E[X] \cdot E[Y]$

Доказательство:

1. По определению $E[X] = \sum_{x \in X(\Omega)} x \cdot P(X = x)$, $E[Y] = \sum_{y \in Y(\Omega)} y \cdot P(Y = y)$.

2. Тогда $E[X] \cdot E[Y] = (\sum_{x \in X(\Omega)} x \cdot P(X = x)) \cdot (\sum_{y \in Y(\Omega)} y \cdot P(Y = y)) = \sum_{(x,y) \in X(\Omega) \times Y(\Omega)} xyP(X = x)P(Y = y) = \sum_{(x,y) \in X(\Omega) \times Y(\Omega)} xyP((X = x) \cap (Y = y))$.

Положим $xy = z$, тогда это равно $\sum_{z \in (XY)(\Omega)} z \cdot P(X \cdot Y = z) = E[X \cdot Y]$

Ч.Т.Д.

Следствие: Если случайные величины X, Y независимы, то $D[X + Y] = D[X] + D[Y]$.

Доказательство:

По определению $D[X] = E[X^2] - (E[X])^2$, тогда $D[X+Y] = E[(X+Y)^2] - (E[X+Y])^2 = E[X^2 + 2XY + Y^2] - (E[X])^2 - 2E[X]E[Y] - (E[Y])^2 = E[X^2] + 2E[XY] + E[Y^2] - (E[X])^2 - 2E[XY] - (E[Y])^2 = E[X^2] + E[Y^2] - (E[X])^2 - (E[Y])^2 = D[X] + D[Y]$

(пользовались тем фактом, что $E[X+Y] = E[X] + E[Y]$)

Ч.Т.Д.

Упражнение: Если X_1, X_2, \dots, X_n - случайные независимые величины, то

- $E[X_1 \cdot X_2 \cdot \dots \cdot X_n] = E[X_1] \cdot \dots \cdot E[X_n]$
- $D[X_1 + X_2 + \dots + X_n] = D[X_1] + D[X_2] + \dots + D[X_n]$

5.3 Оценка биномиальных коэффициентов

(добро пожаловать в матан)

Вот представим себе, что мы подбрасываем монетку n раз. Посчитаем $P(\text{выпало } \frac{n}{2} \text{ орлов})$, что очевидно равно $\binom{\frac{n}{2}}{2^n}$. Вот мы хотим что-то понять про эту вероятность, но для этого нам очень надо уметь оценивать биномиальный коэффициент из числителя.

Давайте выпишем пару очевидных оценок: $\frac{2^n}{n+1} \leq \binom{\frac{n}{2}}{2^n} \leq \sum_{i=0}^n \binom{n}{i} = 2^n$.

Левая оценка верна просто потому что это сумма положительных чисел и туда входит наш биномиальный коэффициент, поэтому очевидно, а правая оценка получена путём рассмотрения строки в треугольнике Паскаля и замечанием того, что наш рассматриваемый биномиальный коэффициент стоит ровно посередине нашей строки Паскаля, поэтому он среди всех в этой строке самый большой. Всего элементов в строке $n+1$, ну и сумма их равна 2^n , поэтому это не что иное, как неравенство о среднем арифметическом и максимумом из всех чисел.

Давайте взглянем теперь на наш биномиальный коэффициент по-другому. $\binom{\frac{n}{2}}{2^n} = \frac{n!}{(\frac{n}{2})! \cdot (\frac{n}{2})!}$. Именно поэтому мы хотим оценить факториалы.

Лемма (Формула Стирлинга): $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$

Вернёмся обратно к нашей вероятности и попробуем посмотреть, что с ней творится на $+\infty$.

1. Вспомним fun fact с матана. $f(n) \sim g(n) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$.

2. Тогда требуется доказать, что $\lim_{n \rightarrow \infty} \frac{n!}{(\frac{n}{2})! \cdot (\frac{n}{2})! \cdot 2^n} = \lim_{n \rightarrow \infty} \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{2^n \cdot \sqrt{\pi n} \cdot \sqrt{\pi n} \cdot \left(\frac{n}{2e}\right)^{\frac{n}{2}} \cdot \left(\frac{n}{2e}\right)^{\frac{n}{2}}} = \lim_{n \rightarrow \infty} \frac{1}{\sqrt{\frac{\pi n}{2}}} = 0$.

Получаем, что наша вероятность при достаточно больших n стремится к 0.

Теперь сформулируем ещё парочку лемм.

Лемма: a) $\binom{n}{k}^k \leq \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$, б) $\sum_{i=0}^k \binom{n}{i} \leq \left(\frac{en}{k}\right)^k$

Доказательство:

1. Докажем сначала пункт а) и его правую оценку.

Мини лемма: $\frac{a}{b} \leq \frac{a-1}{b-1}$ при $a \geq b \geq 1$.

Доказательство: Перемножим крест на крест и получим, что $a(b-1) \leq b(a-1) \implies a \geq b$, что верно. **Ч.Т.Д.**

Тогда так как $\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k(k-1)\dots(k-k+1)}$, то в силу мини леммы получаем, что $\binom{n}{k} \geq \left(\frac{n}{k}\right)^k$

2. Теперь докажем в левую оценку пункта а), однако для этого понадобится сначала доказать пункт б).

Доказательство:

Заметим, что $\sum_{i=0}^k \binom{n}{i} \leq \sum_{i=0}^n \binom{n}{i} \cdot \frac{t^i}{t^k}$, где $t \in (0,1]$. Тогда можно вынести $\frac{1}{t^k}$ за сумму и получим $\frac{1}{t^k} \cdot (1+t)^n$

Пусть теперь $t = \frac{k}{n}$, тогда $\frac{(1+t)^n}{t^k} = \frac{n^k}{k^k} \cdot (1 + \frac{k}{n})^n$. Ну и теперь ещё один fun fact с матана, что $e^k \geq (1 + \frac{k}{n})^n$, поэтому в итоге $\sum_{i=0}^k \binom{n}{i} \leq \frac{n^k}{k^k} \cdot e^k = \left(\frac{en}{k}\right)^k$

Ч.Т.Д.

3. Ну и для полного завершения доказательства осталось заметить, что $\binom{n}{k} \leq \sum_{i=0}^k \binom{n}{i} \leq (\frac{en}{k})^k$, тем самым мы доказали и левую оценку пункта а).

Ч.Т.Д.

5.4 Неравенство Чернова

(сначала может показаться, что матан отступил, но не тут-то было)

Пусть у нас есть монетка, в которой выпадает орёл с вероятностью p и есть случайные независимые величины X_1, X_2, \dots, X_n и $P(X_i = 0) = 1 - p$, а $P(X_i = 1) = p$. Тогда случайная величина $X = \sum_{i=1}^n X_i$ - число выпадения орлов.

Пусть также наше вероятностное пространство - это $\Omega = \{0,1\}^n$, то есть 0 - это решка и 1 - это орёл. Тогда обозначим за a_1, a_2, \dots, a_n результаты подбрасывания монетки n раз, то есть $a_i \in \Omega$ - это результат в i -ом броске.

$$\text{Тогда } P(a_1 \cap a_2 \cap \dots \cap a_n) = p \left(\sum_{i=1}^n a_i \right) \left(1 - p \right)^{(n - \sum_{i=1}^n a_i)}$$

Перед формулировкой и доказательством самого неравенства Чернова нужен..., угадайте кто... правильно -fun fact с матана.

Лемма: $e^x \geq 1 + x$ (не думаю, что на колке попросят доказывать, но тут просто надо перенести всё в одну сторону и исследовать функцию на монотонность и точки экстремума, взяв производную, и всё получится)

Теорема (Неравенство Чернова):

$$\forall \varepsilon \in (0,1) \quad P(X \geq (1 + \varepsilon)pn) \leq e^{-\frac{\varepsilon^2 np}{3}}$$

Неравенство Чернова утверждает, что вероятность того, что число выпавших орлов отклонится от математического ожидания числа орлов не менее, чем в эпсилон раз, не более $e^{-\frac{\varepsilon^2 np}{3}}$

Доказательство:

1. Рассмотрим случайную величину $Y = e^{tX}$, где $t > 0$. Тогда перепишем неравенство Чернова в терминах случайной величины Y и запишем неравенство Маркова (если хз, что оно, то прочитайте бля уже лекцию) для левой части неравенства.

$$P(Y \geq e^{t(1+\varepsilon)pn}) \leq \frac{E[e^{tX}]}{e^{t(1+\varepsilon)pn}}$$

2. Посчитаем мат. ожидание по определению для каждой возможной последовательности орлов и решек: $E[e^{tX}] = \sum_{k=0}^n \binom{n}{k} \cdot p^k (1-p)^{n-k} \cdot e^{tk} = (e^t p + (1-p))^n$. Свернём по битону Ньютона.

$$3. \text{ Значит, } \frac{E[e^{tX}]}{e^{t(1+\varepsilon)pn}} = \frac{(e^t p + (1-p))^n}{e^{t(1+\varepsilon)pn}}$$

Оценим числитель: $(e^t p + (1-p))^n = (1 + \frac{pn(e^t - 1)}{n})^n \leq e^{pn(e^t - 1)}$

$$\text{Итого, } E[e^{tX}] \leq \frac{e^{pn(e^t - 1)}}{e^{t(1+\varepsilon)pn}} = e^{np(e^t - 1 - t(1+\varepsilon))}$$

4. Мы хотим сделать оценку как можно более точную, поэтому надо подобрать такой t , чтобы $e^{np(e^t - 1 - t(1+\varepsilon))}$ был минимален.

Давайте рассмотрим функцию $g(t) = e^t - 1 - t(1+\varepsilon) \implies g'(t) = e^t - (1+\varepsilon) \implies t = \ln(1+\varepsilon)$. Можно убедиться, что полученная точка - это точка минимума. Подставим вместо t её и получим, что $e^{np(\varepsilon - \ln(1+\varepsilon)(1+\varepsilon))}$.

$$\text{Так как } \ln(1+\varepsilon) = \varepsilon - \frac{\varepsilon^2}{2} + \frac{\varepsilon^3}{3} - \dots = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\varepsilon^k}{k}$$

$$\text{Тогда } (1+\varepsilon)\ln(1+\varepsilon) = (1+\varepsilon)(\varepsilon - \frac{\varepsilon^2}{2} + \frac{\varepsilon^3}{3} - \dots) = \varepsilon + \frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{6} + \frac{\varepsilon^4}{12} - \frac{\varepsilon^5}{20} + \dots = \varepsilon + \sum_{k=2}^{\infty} (-1)^k \frac{\varepsilon^k}{k(k-1)} \geq \varepsilon + \frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{6} \geq \varepsilon + \frac{\varepsilon^2}{2} - \frac{\varepsilon^2}{6} \geq \varepsilon + \frac{\varepsilon^2}{3}$$

5. Окончательно получаем, что $\frac{E[e^{tX}]}{e^{t(1+\varepsilon)p^n}} \leq e^{np(\varepsilon - \ln(1+\varepsilon)(1+\varepsilon))} \leq e^{np(\varepsilon - (\varepsilon + \frac{\varepsilon^2}{3}))} = e^{-\frac{\varepsilon^2 np}{3}}$

Ч.Т.Д.

6 Лекция 18

([Никита](#))

6.1 Производящие функции

На данной лекции начнём с весьма формального взгляда на производящие функции.

$F = (f_1, f_2, f_3, \dots)$, $f_i \in \mathbb{C}$ - некоторая бесконечная последовательность комплексных чисел.

Тогда мы ей можем сопоставить "функцию" (строго говоря, по определению это функцией являться не будет):

$$F(x) = f_0 + f_1x + f_2x^2 + f_3x^3 + \dots = \sum_{k=0}^{+\infty} f_k x^k.$$

Стоит воспринимать эту запись как некоторый формальный (степенной) ряд, то есть мы не собираемся ничего туда подставлять (по крайней мере на данный момент). Можно воспринимать x, x^2, x^3, \dots просто как "картинки".

Тогда $F(x) = \sum_{k=0}^{+\infty} f_k x^k$ называют производящей функцией для последовательности F .

Теперь мы хотим определить некоторые арифметические операции для производящих функций:

- Сложение, $+$: $F(x) = \sum_{k=0}^{+\infty} f_k x^k$, $G(x) = \sum_{k=0}^{+\infty} g_k x^k \rightarrow (F+G)(x) = \sum_{k=0}^{+\infty} (f_k + g_k) x^k$
- Умножение на скаляр: $F(x), c \in \mathbb{C} \rightarrow (cF)(x) = \sum_{k=0}^{+\infty} (cf_k) x^k$

Вместе с операциями сложения и умножения множество производящих функций (обозначается как $\mathbb{C}[[x]]$) образует векторное пространство над \mathbb{C} . Проверка аксиом векторного пространства предоставляется читателю в качестве упражнения.

Также покажем как происходит умножение производящих функций:

Интуитивно (это важно) умножение можно выполнить таким образом:

$(F \cdot G)(x) = (f_0 + f_1x + f_2x^2 + \dots)(g_0 + g_1x + g_2x^2 + \dots) = f_0g_0 + (f_1g_0 + f_0g_1)x + (f_2g_0 + f_1g_1 + f_0g_2)x^2 + \dots + (\sum_{k=0}^n f_k g_{n-k})x^n + \dots$. Однако это лишь некоторый способ понять суть умножения, а так оно формально задаётся определением:

$$(F \cdot G)(x) = F(x) \cdot G(x) = H(x), \text{ где } h_n = \sum_{k=0}^n f_k g_{n-k}$$

Теперь мы можем показать, что выполняются некоторые привычные свойства умножения.

Утверждение. $A(x) \cdot (B(x) \cdot C(x)) = (A(x) \cdot B(x)) \cdot C(x)$.

Примечание. $F(x) = G(x) \Leftrightarrow f_i = g_i \forall i \in \mathbb{N} \cup \{0\}$

Доказательство:

$$A(x) \cdot B(x) = D(x). \text{ Посмотрим на коэффициент при } x^k : \sum_{l=0}^k a_l b_{k-l} = \sum_{\substack{i+j=0 \\ i,j \geq 0}} a_i b_j.$$

$$D(x) \cdot C(x). \text{ Посмотрим на коэффициент при } x^n : \sum_{k=0}^n d_k c_{n-k} = \sum_{k=0}^n \sum_{\substack{i+j=0 \\ i,j \geq 0}} a_i b_j c_{n-k} = \sum_{\substack{i+j+t=n \\ i,j,t \geq 0}} a_i b_j c_t.$$

Последний переход следует из того, что сумма всех коэффициентов равна n , то есть мы просто рассматриваем произведение всех коэффициентов таких, чтобы их сумма давала n . Проделав такие же действия для левой части равенства, также получим инвариантное относительно n (по сумме индексов) выражение. Значит, ассоциативность умножения верна ■

Примечание. Мы не отметили это ранее, но умножение производящих функций коммутативно ровно потому, что коммутативно умножение комплексных чисел (см. определение умножения).

Давайте выпишем свойства умножения:

1. $A(x) \cdot B(x) = B(x) \cdot A(x)$ - коммутативность.
2. $A(x) \cdot (B(x) \cdot C(x)) = (A(x) \cdot B(x)) \cdot C(x)$ - ассоциативность.

3. $A(x) \cdot (B(x) + C(x)) = A(x) \cdot B(x) + A(x) \cdot C(x)$ - дистрибутивность.

Доказательство:

Посмотрим на коэффициент при x^n у выражения слева: $\sum_{k=0}^n a_k(b_{n-k} + c_{n-k}) = \sum_{k=0}^n a_k b_{n-k} + \sum_{k=0}^n a_k c_{n-k} = A(x)C(x) + B(x)C(x)$.

4. $\exists 1 = (1, 0, 0, \dots) = 1 + 0x + 0x^2 + \dots$

$$1 \cdot A(x) = A(x)$$

Утверждение. Введём определение константы: $c = (c, 0, 0, \dots)$, $c(x) = c + 0x + 0x^2 + \dots$ Тогда $c(x) \cdot A(x) = (cA)(x)$

Доказательство:

Посмотрим на коэффициент при x^n у выражения слева: $\sum_{k=0}^n c_k a_{n-k} = ca_n$, т.к. все c_k кроме c_0 равны 0.

5. Обратный элемент не всегда существует.

Определение. Пусть $A(x)$ - производящая функция. Тогда $B(x)$ - обратная к $A(x)$, если $A(x) \cdot B(x) = 1$ ($B(x) = A(x)^{-1}$)

Примеры:

$A(x) = 1 + x + x^2 + x^3 + \dots$ - обратима. Для неё существует обратный: $1 - x$. $A(x)(1 - x) = 1 + 0x + 0x^2 + \dots = 1$
 $B(x) = x$ - необратима, т.к. $xC(x) = 0 + c_0x + c_1x^2 + \dots \neq 1$

Будем обозначать обратный таким образом: $A(x)^{-1} = \frac{1}{A(x)}$.

Запишем несколько свойств, которые позже будут доказаны на семинарах:

$$1. \frac{1}{A(x)} \cdot \frac{1}{B(x)} = \frac{1}{A(x)B(x)}$$

$$2. \frac{A(x)}{B(x)} + \frac{C(x)}{D(x)} = \frac{A(x)D(x) + B(x)C(x)}{B(x)D(x)}$$

Поясним, что по определению $\frac{A(x)}{B(x)}$ значит $A(x) \cdot B(x)^{-1}$.

6.2 Дифференцирование

По **определению** производная к производящей функции $F = f_0 + f_1x + f_2x^2 + \dots$ это:

$$F'(x) = f_1 + 2f_2x^2 + 3f_3x^3 + \dots = \sum_{k=1}^{+\infty} kf_k x^{k-1}$$

Свойства производных:

$$1. (F(x) + G(x))' = F'(x) + G'(x)$$

Доказательство:

Посмотрим на коэффициент при x^{k-1} у выражения слева: $k(f_k + g_k) = kf_k + kg_k$.

$$2. (cF(x))' = cF'(x) = (cF'(x))(x)$$

Доказательство:

Посмотрим на коэффициент при x^{k-1} у выражения слева: $k \cdot cf_k = ckf_k$.

$$3. \text{Правило Лейбница: } (F(x)G(x))' = F'(x)G(x) + F(x)G'(x).$$

Доказательство:

Посмотрим на коэффициент при x^{n-1} у выражения $(F(x)G(x))'$: при x^n в $F(x)G(x)$ коэффициент равен $\sum_{k=0}^n f_k g_{n-k}$, значит при x^{n-1} у $(F(x)G(x))'$ он равен $n(f_0g_n + f_1g_{n-1} + \dots + f_ng_0)$.

Теперь посмотрим на коэффициент при x^{n-1} у выражения $F'(x)G(x)$: $\sum_{k=0}^{n-1} (k+1)f_{k+1}g_{n-1-k} = f_1g_{n-1} + 2f_2g_{n-2} + 3f_3g_{n-3} + \dots + nf_ng_0$.

И на коэффициент при x^{n-1} у выражения $F(x)G'(x)$ (он почти такой же как у предыдущего выражения): $g_1f_{n-1} + 2g_2f_{n-2} + 3g_3f_{n-3} + \dots + ng_nf_0$.

Тогда рассмотрим сумму коэффициентов при x^{n-1} у выражений $F'(x)G(x)$ и $F(x)G'(x)$: любой $f_i g_{n-i}$ входит в сумму два раза с коэф-ми i и $n - i$, которые в сумме дадут n . Тогда коэффициент у данной суммы как раз и будет $nf_0g_n + nf_1g_{n-1} + \dots + nf_ng_0 = n(f_0g_n + f_1g_{n-1} + \dots + f_ng_0)$

$$4. \left(\frac{1}{F(x)}\right)' = -\frac{F'(x)}{F^2(x)} - \text{будет обговорено на семинаре}$$

$$5. \left(\frac{A(x)}{B(x)}\right)' = \frac{A'(x)B(x) - A(x)B'(x)}{B^2(x)}$$

Сводится к (3) и (4), т.к. $\left(\frac{A(x)}{B(x)}\right)' = (A(x) \cdot \frac{1}{B(x)})'$

6.3 Пример применения

Задача будет заключаться в вычислении $\sum_{i=0}^n i^2$ с помощью производящих функций. Т.е. по факту мы хотим найти производящую функцию для последовательности $(0^2, 0^2 + 1^2, 0^2 + 1^2 + 2^2, \dots)$.

Возьмём $F(x) : (1, 1, 1, \dots) \rightarrow 1 + x^2 + x^3 + \dots$; $F(x) = \frac{1}{1-x}$

$F'(x) : (1, 2, 3, \dots); F'(x) = (\frac{1}{1-x})' = \frac{1}{(1-x)^2}$

Давайте получим это же но с 0, домножив на x :

$xF'(x) : (0, 1, 2, 3, \dots); xF'(x) = \frac{x}{(1-x)^2}$

$(xF'(x))' : (1^2, 2^2, 3^2, \dots); (xF'(x))' = (\frac{x}{(1-x)^2})' = \frac{1+x}{(1-x)^3}$

Давайте снова получим это же но с 0, домножив на x :

$x(xF'(x))' : (0^2, 1^2, 2^2, 3^2, \dots); x(xF'(x))' = \frac{x(1+x)}{(1-x)^3}$

Утверждение. Пусть $A(x) \rightarrow (a_0, a_1, a_2, a_3, \dots)$

$S = (a_0, a_0 + a_1, a_0 + a_1 + a_2, \dots, \sum_{k=0}^n a_k, \dots)$. Тогда $S(x) = \frac{A(x)}{1-x}$.

Доказательство:

Просто по определению $\frac{A(x)}{1-x} = A(x)(1 + x + x^2 + \dots) = S(x)$

Поэтому производящая функция для $(0^2, 0^2 + 1^2, 0^2 + 1^2 + 2^2, \dots) \rightarrow \frac{x(xF'(x))'}{(1-x)} = \frac{x(1+x)}{(1-x)^4}$

$G(x) = \frac{x(1+x)}{(1-x)^4}, g_n = \sum_{i=0}^n i^2$

Теперь наша задача понять, как "вытащить" из производящей функции её коэффициент. Хотя мы договорились пока ничего не подставлять в нашу производящую функцию, давайте определим подстановку 0:

$A(x) = a_0 + a_1 x + a_2 x^2 + \dots$, тогда $A(0) = a_0$.

Запишем несколько очевидных свойств:

1. $(A + B)(0) = A(0) + B(0)$
2. $(A \cdot B)(0) = A(0) \cdot B(0)$
3. $(cA)(0) = cA(0)$

Утверждение. $A \rightarrow (a_0, a_1, a_2, \dots)$. Тогда $a_n = \left. \frac{A^{(n)}(x)}{n!} \right|_{x=0}$ Несложно проверить, взяв производную n раз.

Разложим $G(x)$ на сумму двух дробей $\frac{x(1+x)}{(1-x)^4} = \frac{x^2}{(1-x)^4} + \frac{x}{(1-x)^4}$. Но искать n -ые производные данных дробей тоже не просто, поэтому можем ввести вспомогательную производящую функцию $H(x) = \frac{1}{(1-x)^4}$. Понятно, что умножение на x^2 и x делает сдвиг коэффициентов $H(x)$ на 2 и на 1 вправо соответственно, поэтому коэффициенты h_{n-2} и h_{n-1} будут n -ми коэффициентами данных дробей соответственно. Тогда посчитаем производную в общем виде для $\frac{1}{(1-x)^4}$:

$H'(x) = \frac{5}{(1-x)^5}, H''(x) = \frac{4 \cdot 5}{(1-x)^6}, H'''(x) = \frac{4 \cdot 5 \cdot 6}{(1-x)^7}, \dots$ по индукции несложно проверяется, что $H^{(n)}(x) = \frac{4 \cdot 5 \cdot 6 \cdots (n+3)}{(1-x)^{n+4}} = \frac{(n+3)!}{6(1-x)^{n+4}}$.

Тогда $h_n = \left. \frac{(n+3)!}{n! \cdot 6(1-x)^4} \right|_{x=0} = \frac{(n+1)(n+2)(n+3)}{6}$

Ответом будет $g_n = \sum_{i=0}^n i^2 = h_{n-2} + h_{n-1} = \frac{(n-1)n(n+1)}{6} + \frac{n(n+1)(n+2)}{6} = \frac{n(n+1)(2n+1)}{6}$

6.4 Ещё один пример применения (бонусный)

Пусть $A(x)$ - производящая функция для неупорядоченных выборок из S (то есть a_n = количество способов выбрать n элементов из S без учёта порядка). Аналогично $B(x)$ - такая же функция, но для множества T . Будем также считать, что $S \cap T = \emptyset$

Неформальное утверждение. Производящей функцией для количества выборок из $S \cup T$ является функция $A(x) \cdot B(x)$.

Пример про салат (всё-таки прикладная математика)

Сразу отметим, что все овощи идентичны. Мы хотим сделать салат из огурцов (можно добавлять сколько угодно), помидоров (можно добавить или 0 шт., или 2 шт.) и авокадо (можно добавить не более 3 шт.). Тогда производящая функция для неупорядоченных выборок огурцов - $C(x) = 1 + x + x^2 + x^3 + \dots$, помидоров - $T(x) = 1 + x^2$, авокадо - $A(x) = 1 + x + x^2 + x^3$.

Тогда производящая функция для объединения это $C(x)T(x)A(x) = (1 + x + x^2 + x^3 + \dots)(1 + x^2)(1 + x + x^2 + x^3) = \frac{(1+x^2)(1+x+x^2+x^3)}{1-x} = \frac{(1+x^2)(1-x^4)}{(1-x)^2}$

7 Лекция 19

(Андрей)

7.1 Неупорядоченные выборки

Пусть имеем n различных объектов, из которых мы неупорядоченно хотим выбрать k штук. Конечно, мы могли бы это сделать, пользуясь обычной комбинаторикой, но иногда жизнь несправедлива, поэтому применим соображения из производящих функций.

Рассмотрим n -элементный набор $S = \{a_1, a_2, \dots, a_n\}$ который представим в виде объединения n одноэлементных множеств, то есть $S = \{a_1\} \cup \{a_2\} \cup \dots \cup \{a_n\}$.

Рассмотрим произвольный элемент a_i . Понятно, что для него производящая функция в смысле нашей задачи будет иметь вид: $1 + x$. Далее, из прошлой лекции мы знаем, что производящая функция для объединения n объектов будет равна произведению производящих функций каждого из этих объектов. Поэтому искомая функция будем иметь вид:

$$(1+x)(1+x)\dots(1+x) = (1+x)^n = \sum_{i=0}^n C_n^i \cdot x^i$$

Таким образом, мы получили функцию вида: $f(x) = f_0 + f_1x + f_2x^2 + \dots + f_nx^n$, где коэффициент f_i обозначает количество способов выбрать неупорядоченно k элементов из n .

7.2 Бином Ньютона

Давайте обобщим формулу бинома Ньютона для целых коэффициентов и получим:

$$(1+x)^{-n} = \sum_{i=0}^{\infty} \binom{-n}{i} \cdot x^i$$

Давайте поясним, чему равен коэффициент $\binom{-n}{k}$.

$$\binom{-n}{k} = \frac{(-n)(-n-1)\dots(-n-k+1)}{k!} = (-1)^k \cdot \frac{n(n+1)\dots(n+k-1)}{k!} = (-1)^k \cdot \frac{(n+k-1)!}{(n-1)! \cdot k!} = (-1)^k \cdot \binom{n+k-1}{k}$$

Отсюда предыдущая формула переписывается в виде:

$$(1+x)^{-n} = \sum_{i=0}^{\infty} \binom{-n}{i} \cdot x^i = \sum_{i=0}^{\infty} \binom{n+i-1}{i} \cdot x^i$$

На самом деле, можно пойти дальше и определить биномиальный коэффициент $\binom{\alpha}{k}$, где $\alpha \in \mathbb{R}, k \in \mathbb{N}$.

$$\binom{\alpha}{k} = \frac{\alpha(\alpha+1)\dots(\alpha-k+1)}{k!}$$

Отдельно стоит отметить, что $\binom{\alpha}{0} = 1$.

Пример:

Рассмотрим равенство $(1+x)^k \cdot (1+x)^l = (1+x)^{k+l}$, $k, l \in \mathbb{N}$. Оно очевидно, если мы рассматриваем его как равенство многочленов, но давайте взглянем на него на как на равенство производящих функций. Перепишем наше равенство:

$$\sum_{i=0}^{\infty} \binom{k}{i} x^i \cdot \sum_{j=0}^{\infty} \binom{l}{j} x^j = \sum_{n=0}^{\infty} \binom{k+l}{n} x^n.$$

При x^n стоит коэффициент:

$$\sum_{\substack{i+j=n \\ i,j \geq 0}} \binom{k}{i} \binom{l}{j} = \binom{k+l}{n}$$

Справа и слева мы получили многочлены от k и l , при этом n фиксировано. Заметим, что эти многочлены равны в бесконечном количестве точек. Далее, зафиксируем $l \in \mathbb{N} \Rightarrow$ левая часть равна правой части $\forall k \in \mathbb{R}$. Затем зафиксируем $k \in \mathbb{R}$ и получим, что левая часть равна правой части $\forall l \in \mathbb{R} \Rightarrow$ эти многочлены равны $\forall k, l \in \mathbb{R}$.

7.3 Линейные рекуррентные соотношения с постоянными коэффициентами

Найдем явную формулу для n -го числа Фибоначчи, используя производящие функции.

Числа Фибоначчи задаются следующим рекуррентным соотношением:

$$F_{n+2} = F_{n+1} + F_n$$

причем $F_0 = F_1 = 1$.

Рассмотрим производящую функцию для последовательности Фибоначчи:

$$F(x) = F_0 + F_1 \cdot x + F_2 \cdot x^2 + \dots + F_n \cdot x^n + \dots$$

Умножим $F(x)$ на x , а потом на x^2 . Получаем:

$$xF(x) = F_0 \cdot x + F_1 \cdot x^2 + F_2 \cdot x^3 + F_3 \cdot x^4 + \dots + F_n \cdot x^{n+1} + \dots$$

$$x^2F(x) = F_0 \cdot x^2 + F_1 \cdot x^3 + F_2 \cdot x^4 + F_3 \cdot x^5 + \dots + F_n \cdot x^{n+2} + \dots$$

Сложим получившиеся две функции:

$$(x + x^2) \cdot F(x) = F_0 \cdot x + (F_0 + F_1) \cdot x^2 + (F_1 + F_2) \cdot x^3 + \dots + (F_{n-1} + F_n) \cdot x^{n+1} + \dots$$

Заметим, что каждый коэффициент этой функции тоже является числом Фибоначчи, поэтому перепишем эту функцию в таком виде:

$$(x + x^2) \cdot F(x) = F_1 \cdot x + (F_2) \cdot x^2 + (F_3) \cdot x^3 + \dots + (F_{n+1}) \cdot x^{n+1} + \dots = F(x) - F_0 = F(x) - 1$$

Теперь можем получить явный вид $F(x)$:

$$F(x) = \frac{1}{1 - x - x^2} = \frac{-1}{(x - x_1)(x - x_2)}$$

$$x_1 = \frac{-1 + \sqrt{5}}{2}, \quad x_2 = \frac{-1 - \sqrt{5}}{2}$$

Преобразуем наше выражение:

$$\begin{aligned} \frac{-1}{(x - x_1)(x - x_2)} &= \left(\frac{1}{x - x_1} - \frac{1}{x - x_2} \right) \cdot \frac{1}{x_2 - x_1} = \frac{-1}{\sqrt{5}} \cdot \left(\frac{1}{x - x_1} - \frac{1}{x - x_2} \right) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1}{x - x_2} - \frac{1}{x - x_1} \right) = \\ &= \frac{1}{\sqrt{5}} \cdot \left(\frac{1}{-x_2 \cdot (1 - \frac{x}{x_2})} - \frac{1}{-x_1 \cdot (1 - \frac{x}{x_1})} \right) = \frac{1}{x_1 \cdot \sqrt{5}} \cdot (1 - \frac{x}{x_1})^{-1} - \frac{1}{x_2 \cdot \sqrt{5}} \cdot (1 - \frac{x}{x_2})^{-1} = \\ &= \frac{1}{x_1 \cdot \sqrt{5}} \cdot (1 + \frac{x}{x_1} + \frac{x^2}{x_1^2} + \dots) - \frac{1}{x_2 \cdot \sqrt{5}} \cdot (1 + \frac{x}{x_2} + \frac{x^2}{x_2^2} + \dots) = \dots + x^n \cdot \left(\frac{1}{x_1^{n+1} \cdot \sqrt{5}} - \frac{1}{x_2^{n+1} \cdot \sqrt{5}} \right) \end{aligned}$$

Итого: $F_n = \frac{(-1)^{n+1}}{\sqrt{5}} \cdot (x_2^{n+1} - x_1^{n+1}) = \frac{1}{\sqrt{5}} \cdot \left(\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1} \right)$. Победа, получили формулу Бине!

7.4 Теорема о рекуррентном соотношении

Пусть имеем некоторую рекуррентную последовательность $\{a_i\}_{i=0}^{\infty}$ и пусть для некоторого $k \in \mathbb{N}$ выполняется, что a_j член выражается через k предыдущих с постоянными коэффициентами. То есть:

$$a_{n+k} = c_1 a_{n+k-1} + c_2 a_{n+k-2} + \dots + c_k a_n$$

Тогда если рассмотрим производящую функцию $A(x) = a_0 + a_1 x + a_2 x^2 + \dots$

То $A(x) = \frac{P(x)}{Q(x)}$, где $P(x), Q(x) \in \mathbb{C}(x)$ и $\deg Q = k$, $\deg P \leq k - 1$.

Доказательство:

Рассмотрим произведение $A(x) \cdot (c_1x + c_2x^2 + \dots + c_kx^k)$:

$$c_1x \cdot A(x) = c_1a_0x + c_1a_2x^2 + \dots + c_1a_{n+k-1}x^{n+k} + \dots$$

$$c_1x^2 \cdot A(x) = c_1a_0x^2 + c_1a_2x^3 + \dots + c_1a_{n+k-2}x^{n+k} + \dots$$

Делаем так k раз и получаем:

$$c_kx^k \cdot A(x) = c_ka_0x^k + \dots + c_ka_nx^{n+k} + \dots$$

Просуммируем все эти выражения и получим:

$$A(x)(c_1x + c_2x^2 + \dots + c_kx^k) = (\text{какой-то многочлен } R(x)) + a_kx^k + a_{k+1}x^{k+1} + \dots = A(x) + P(x)$$

$P(x)$ - тоже некоторый многочлен, причем $\deg P(x) \leq k - 1$.

Теперь вспомним, что рассматривали произведение $A(x) \cdot (c_1x + c_2x^2 + \dots + c_kx^k)$. Мы получили, что оно равно $A(x) + P(x)$, откуда $P(x) = A(x) \cdot (-1 + c_1x + c_2x^2 + \dots + c_kx^k)$. Обозначим $(-1 + c_1x + c_2x^2 + \dots + c_kx^k)$ за $Q(x)$ и получим, что $A(x) = \frac{P(x)}{Q(x)}$.

Ч.Т.Д.

7.5 Нахождение общей формулы для a_n

Теперь мы получили, что $A(x) = \frac{P(x)}{Q(x)} = \frac{P(x)}{(1-q_1x)^{r_1} \dots (1-q_nx)^{r_s}} = \dots + \frac{A}{(1-q_i \cdot x)^m} + \dots$ (разложение в сумму простых дробей (A - это какой-то коэффициент)). Заметим, что слагаемое вида:

$$\frac{A}{(1-q_i \cdot x)^m}$$

в точности равно:

$$A \cdot \sum_{k=0}^{\infty} \binom{-m}{k} \cdot (-q_i \cdot x)^k$$

Тогда чтобы получить коэффициент при x^n у производящей функции, мы пройдемся по всем элементарным дробям и соберем в сумму все коэффициенты при x^n . Причем итоговый коэффициент будем иметь вид:

$$a_n = q_1^n \cdot p_1(n) + \dots + q_t^n \cdot p_t(n)$$

$(p_i(n)$ - какой-то многочлен от n , который возникает из-за биномиального коэффициента при q_i^n)

Выражение вида $p_i(n) \cdot q_i^n$ называют квазимногочленом.

7.6 Числа Каталана

Задача о правильной скобочной последовательности

Я буду обозначать правильную скобочную последовательность как **ПСП**.

ПСП - это такая последовательность, которая получается за конечное число шагов по следующим правилам:

1. () - ПСП.
2. Если P - ПСП, то (P) - тоже ПСП.
3. Если P_1 и P_2 - ПСП, то $P_1 + P_2$ - ПСП (под + подразумевается конкатенация).

Обозначим за C_n - число правильных скобочных последовательностей длины $2n$. Найдем это число, выразив число C_{n+1} через предыдущие.

Для этого рассмотрим первый момент, когда число левых скобок равно числу правых скобок (см. картинку)

Получили, что длина P_1 равна $2k$, а длина P_2 равна $2(n+1) - 2k$. Тогда получаем, что число C_{n+1} выражается следующим образом:

$$C_{n+1} = C_0 \cdot C_n + C_1 \cdot C_{n-1} + C_2 \cdot C_{n-2} + \dots + C_n \cdot C_0 = \sum_{k=0}^n C_k \cdot C_{n-k}$$

При этом C_0 мы полагаем равным 1. Рассмотрим производящую функцию для нашего рекуррентного соотношения:

$$C(x) = C_0 + C_1x + C_2x^2 + \dots$$

Рассмотрим произведение $C(x) \cdot C(x)$:

$$C(x) \cdot C(x) \Rightarrow \text{при } x^n \text{ стоит коэффициент } \sum_{k=0}^n C_k \cdot C_{n-k} = C_{n+1}$$

Тогда умножим $C(x) \cdot C(x)$ на x и при члене x^{n+1} получим коэффициент: $\sum_{k=0}^n C_k \cdot C_{n-k} = C_{n+1}$ (что мы и хотели). Но тогда:

$$x \cdot C(x) \cdot C(x) = C_1x + C_2x^2 + \dots = C(x) - C_0$$

Таким образом, получили соотношение:

$$x \cdot C(x)^2 - C(x) + 1 = 0$$

Решая относительно $C(x)$, получим, что:

$$C(x) = \frac{1 \pm \sqrt{1 - 4x}}{2x} = \frac{1 \pm (1 - 4x)^{1/2}}{2x}$$

Далее, возьмем знак минус и раскроем $(1 - 4x)^{1/2}$ по биному и тогда получим явную формулу для $C(x)$:

$$C(x) = \frac{1 \pm \sqrt{1 - 4x}}{2x} = \frac{1 - (1 - 4x)^{1/2}}{2x}$$

Рассмотрим $1 - (1 - 4x)^{1/2}$:

$$\begin{aligned} 1 - (1 - 4x)^{1/2} &= 1 - \sum_{k=1}^{\infty} \binom{\frac{1}{2}}{k} (-4x)^k = -\sum_{k=1}^{\infty} \frac{\frac{1}{2}(\frac{1}{2}-1)\dots(\frac{1}{2}-k+1)}{k!} (-4x)^k = -\sum_{k=1}^{\infty} \frac{1(-1)(-3)\dots(-(2k-3))}{2^k \cdot k!} \cdot (-4x)^k = \\ &= \sum_{k=1}^{\infty} \frac{1(1)(3)\dots(2k-3)}{2^k \cdot k!} 4^k \cdot x^k = \sum_{k=1}^{\infty} \frac{1 \cdot 1 \cdot 3 \cdot \dots \cdot (2k-3) \cdot 2 \cdot 4 \cdot 6 \cdot \dots \cdot (2k-2)}{2^{k-1} \cdot 1 \cdot 2 \cdot 3 \cdot \dots \cdot (k-1) \cdot 2^k \cdot k!} 4^k \cdot x^k = \sum_{k=1}^{\infty} \frac{(2k-2)!}{(k-1)! \cdot (k-1)!} \cdot \frac{2}{k} \cdot x^k \end{aligned}$$

$$\text{Значит, } \frac{1-(1-4x)^{1/2}}{2x} = \sum_{k=1}^{\infty} \frac{(2k-2)!}{(k-1)!(k-1)!} \cdot \frac{1}{k} \cdot x^{k-1} = \sum_{k=1}^{\infty} \binom{2n}{n} \cdot \frac{1}{n+1} \cdot x^n$$

На данный момент мы, можно сказать, угадали ответ - теперь можно строго доказать эту формулу по индукции.

Утверждение: $C_n = \binom{2n}{n} \cdot \frac{1}{n+1}$, $C_0 = 1$, $C_1 = 1$.

Доказательство:

Введем числа $\tilde{C}_n = \binom{2n}{n} \cdot \frac{1}{n+1}$ (знак тильды просто означает, что мы хотим отличать числа Каталана, полученные из производящих функций и полученные из комбинаторных рассуждений). Докажем, что для них выполняется рекуррентное соотношение:

$$\tilde{C}_n = \sum_{\substack{k+l=n-1 \\ k, l \geq 0}} \tilde{C}_k \tilde{C}_l$$

Пока что непонятно, как это можно хорошо доказать. Чтобы сделать нашу жизнь лучше, рассмотрим следующее равенство (просто воспользуемся тем, как мы определили биномиальный коэффициент с дробным показателем):

$$\binom{\frac{1}{2}}{n} = \frac{(-1)^n}{2^{2n-1}} \tilde{C}_{n-1}$$

Правда, при $n = 0$ правая часть не имеет смысла, и чтобы придать ей смысл, надо положить $\tilde{C}_{-1} = -1/2$. Теперь мы можем вспомнить тождество, которое верно $\forall k, l \in \mathbb{R}$:

$$\sum_{\substack{i+j=n \\ i,j \geq 0}} \binom{k}{i} \binom{l}{j} = \binom{k+l}{n}$$

После этого запишем его для $k = 1 = \frac{1}{2}$, правая часть обратится в нуль, и получится тождество:

$$0 = \tilde{C}_{-1} \cdot \tilde{C}_{n-1} + \tilde{C}_0 \cdot \tilde{C}_{n-2} + \tilde{C}_1 \cdot \tilde{C}_{n-3} + \dots + \tilde{C}_{n-1} \cdot \tilde{C}_{-1}$$

Вспоминая о соглашении $C_{-1} = -\frac{1}{2}$, мы получаем искомое рекуррентное соотношение (только для меньшего на единицу значения n).

Ч.Т.Д.

8 Лекция 20

(Илья)

8.1 Комбинаторные игры(определения и примеры)



Начнём с небольшого примера.

Пример(игра в монетницу)

Игроки Первый и Второй по очереди достают монеты из монетницы, в которой изначально лежат 20 монет, 2 или 3 монеты. За один ход соответственно можно взять 2 или 3 монеты. Проигрывает тот, кто не может сделать ход.

Нужно сказать, есть ли у какого-то игрока выигрышная стратегия, если оба игрока будут играть *оптимальным способом*. (пока не очень понятно, как это с формальной точки зрения, но потом станет понятно)

Ответ: Выигрывает Второй.

Решение:

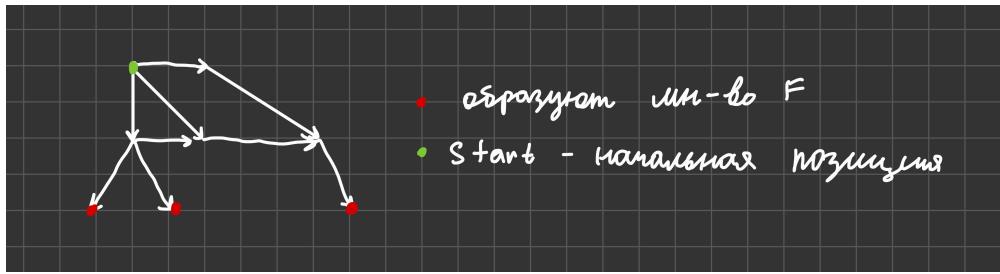
1. Приведём некую стратегию для второго игрока. Заметим, что 20 - это 4 раза по 5 монет.
2. Если Первый берёт 2 монеты, то Второй должен взять 3 и наоборот. То есть Второй своим ходом дополняет ход Первого до 5. Тогда проделав так 4 раза, Второй заберёт последние монеты и тем самым победит.

Мы хотим формально ввести понятие игры, введя всё, что нам нужно для этого.

Задание комбинаторной игры с 2 игроками(будем называть их Макс и Мин, позже поймёте почему), с полной информацией:

1. Множество позиций S , с указанием, кто ходит в конкретной позиции.
2. Начальная позиция: $start$ и множество позиций F , в которых игра заканчивается.
3. Ходы: $\forall s \in S \rightarrow N(s)$ - множество позиций, куда можно попасть из s . Причём, если $s \notin F$, то $N(S) \neq \emptyset$
4. Выигрыш - некая функция $f : F \rightarrow R$

Заметим, что игра - это по сути ориентированный граф (см. картинку), причём мы договоримся, что граф ацикличен, чтобы игра не могла длиться вечно. Понятно, что есть игры, в которых может так получиться, что в игре могут быть циклы, но такие моменты обычно оговорены правилами (например, игра в шахматах про ничью в случае повтора хода 3 раза подряд). Это некий технический момент, который понадобится нам.



Пример описания игры в шахматы:

Заметим, что нам недостаточно знать расстановку фигур и чей ход для описания позиций, потому что в шахматах есть куча нюансов по типу взятия на проходе, рокировки и тд, и нам важно знать, были ли они или нет, чтобы понять кто ходит при той или иной расстановке фигур, поэтому нам необходимо вести за собой по-хорошему всю историю игры.

Пример описания игры в крестики и нолики:

Тут уже всё попроще, поэтому тут под позицией действительно можно понимать просто расстановку какую-то крестиков и ноликов, которая реально могла встретиться в игре (модуль разности крестиков и ноликов равен 1 или 0)

Поговорим про то, как устроена функция выигрыша: (покажу на примерах, и сразу станет понятно)

- В игре в монетницу функция выигрыша может принимать значения 1 и -1, потому что тут можно или выиграть, или проиграть.
- В шахматах и крестиках-ноликах функция выигрыша может принимать значения -1, 0 и 1, потому что может быть ещё и ничья.

Теперь понятно, почему игроков назвали Макс и Мин, потому что Макс стремится максимизировать выигрыши в игре, а Мин минимизировать, поэтому если значение функции равно 1, то победил Макс, а если -1, то Мин.

Дадим ещё пару определений.

Партия- это набор v_0, v_1, \dots, v_k - таких, что $v_i \in S$ и $(v_i, v_{i+1}) \in E$ (ребра графа) и v_0 - начальная позиция и $v_k \in F$

Стратегия (для Макса) - это функция $g : \text{множество позиций} \rightarrow \text{множество позиций}$, где ходит Макс \rightarrow множество позиций, в которую можно сходить ($s \rightarrow v \in N(s)$)

Стратегия для Мина строится полностью аналогично.

Замечание: В нашем курсе мы будем рассматривать только позиционные стратегии, где стратегия зависит только от текущей позиции, то есть нам не важна история всех позиций до этого, а только с конкретной позиции (заметим, что формально это ничего не нарушает, так как у нас само понятие конкретной позиции может включать в себя уже некую историю игры (см. пример с шахматами выше)).

Мы будем говорить, что стратегия g для игрока Макс (для Мин аналогично) гарантирует выигрыш $c \in R$, если для любой партии v_0, v_1, \dots, v_k , согласованной с этой стратегией g , Макс получит выигрыш не меньше c , то есть $f(v_k) \geq c$

Для Мина будет аналогично, но только $f(v_k) \leq c$

Ещё одно определение...

Число C называется **ценой игры**, если у Макса и у Мина есть стратегии, гарантирующие выигрыш C .

8.2 Теорема о цене игры

Теорема о цене игры: У любой описанной выше игры существует единственная цена.

Доказательство:

1. Сначала докажем единственность, потому что она очевидна, ведь если у игры была не одна цена, а , например c_1 и c_2 и не умоляя обности $c_1 < c_2$, то давайте возьмём стратегию Макса, которая гарантирует выигрыш c_2 , то есть $f(v_k) \geq c_2$, и стратегию Мина, которая гарантирует выигрыш c_1 , то есть $f(v_k) \leq c_1$, откуда $c_1 \geq c_2$ - противоречие!

2. Теперь покажем существование. Давайте введём вспомогательное определение:

Назовём позицию **хорошой**, если игра, начинающаяся с этой позиции, имеет цену. Для каждой хорошей позиции мы выберем в этой позиции тот ход, который согласован со стратегией, гарантирующей эту цену.

Лемма: Если все позиции, в которые можно попасть из данной (быть может, за несколько ходов), хороши (следовательно, в каждой из них выбран ход стратегий), то и данная позиция хороша (следовательно, в ней также выбран ход стратегий).

Что говорит эта лемма про заключительные позиции? По правилам логики хороши все позиции, в которые можно попасть из заключительной (таких позиций просто нет, так как из заключительной позиции попасть никуда нельзя). Но и утверждение леммы тривиально: в заключительных позициях результат уже известен, так что оба игрока его уже обеспечили.

Доказательство:

- Для доказательства леммы осталось рассмотреть случай незаключительной позиции. В ней ходит либо Макс, либо Мин. Рассуждения в обоих случаях аналогичны, разберём подробно только случай, когда в позиции x , принятой за начальную, ходит Макс.
- Пусть y_1, \dots, y_k — все позиции, в которые Макс может попасть по правилам игры. По предположению все эти позиции хороши и потому имеют некоторые цены c_1, \dots, c_k . Более того, мы предполагаем, что в каждой из этих позиций, как и во всех позициях, в которые можно попасть из них, выбран ход стратегий. Что должен делать Макс «при правильной игре»?
- Понятное дело — он должен выбрать позицию с наибольшим c_i (одну из таких, если максимум достигается в нескольких) и пойти туда. И дальше следовать той стратегии для позиции y_i , которая гарантирует ему c_i и которая всегда выбирает ход стратегий (по условию такая существует). Так он обеспечит себе результат не меньше c_i . С другой стороны, в какую бы позицию y_j он ни пошёл, у Мина есть такая стратегия, которая обеспечивает ему результат не больше c_j и всегда выбирает ход стратегий. Заметим, что $c_j \leq c_i$, поскольку c_i было максимальным.
- Повторим это рассуждение более формально. Не ограничивая общности, будем считать, что в начальной позиции x ходит Макс. Докажем, что цена игры с началом в x равна $c = \max(c_1, \dots, c_k)$, где c_i — цена игры с началом в y_i , существующая по предположению индукции. Ходом стратегий в позиции x объявим ход в позицию y_i с наименьшим индексом i , для которой достигается максимум, то есть $c_i = c$.
- По предположению позиция y_i хороша, то есть у Макса есть стратегия, которая выбирает ходы стратегий и гарантирует цену не ниже $c = c_i$. Значит, стратегия Макса, которая выбирает ход стратегий в каждой позиции, начиная с x , гарантирует ему результат не меньше c .
- С другой стороны, стратегия Мина, которая выбирает ходы стратегий в позициях, которые возникают в партии, начинающейся с позиции x , гарантирует ему результат не больше c_i . Действительно, пусть Макс пошёл в y_j . Тогда по предположению у Мина есть стратегия, которая гарантирует Мину результат не больше $c_j \leq c$ (c — максимум) в игре с началом в y_j и всегда выбирает ходы стратегий.
- Здесь существенен выбор ходов стратегий в каждой из позиций, благодаря чему стратегия Мина при игре из начальной позиции x корректно определена.

Теперь можно доказать очень просто и существование.

Пусть есть плохая позиция. Тогда по лемме есть позиция, в которую можно из неё перейти, которая тоже будет плохой (в частности, наша позиция не заключительная). Из этой новой плохой позиции по тем же причинам можно перейти в какую-то плохую позицию, и так далее. Рано или поздно (граф ведь конечный) получится цикл, которого быть не может — противоречие.

Ч.Т.Д.

9 Лекция 21

(Никита)

9.1 N и P позиции (выигрышные и проигрышные)

Беспрестрастная игра — это такая игра, в которой два игрока ходят по очереди и проигрывает тот, кто не может сделать ход (иногда рассматривается, что выигрывает тот, кто не может сделать ход). При этом допустимые ходы зависят только от позиции, а не от того, кто из игроков ходит сейчас. Соответственно есть позиции двух типов: $\{1, -1\}$.

Тогда **N-позиции (выигрышные)** — позиции, в которых выигрывает тот, кто ходит следующим (от слова Next).

P-позиции (проигрышные) - позиции, в которых выигрывает тот, кто ходил предыдущим (от слова Previous).

Понятно, что текущая позиция:

- N-позиция, если есть ход в *P*-позицию.
- P-позиция, если все ходы ведут в *N*-позиции.

Исследовать вершины нужно в соответствии с топологической сортировкой. Данная процедура когда мы идём из проанализированных позиций в ещё не проанализированные называется *Анализом с конца*.

Пример:

Приведём пример несложной игры, в которой мы можем применить анализ с конца.

Правила: за свой ход можно подвинуть ладью на сколько угодно клеток вправо или вверх. Проигрывает тот игрок, который не может сдвинуть ладью. Изначально ладья стоит в левой нижней клетке.

N	N	N	N	N	N	N	N	P
N	N	N	N	N	N	P	N	N
N	N	N	N	N	P	N	N	N
N	N	N	N	P	N	N	N	N
N	N	N	P	N	N	N	N	N
N	N	P	N	N	N	N	N	N
N	P	N	N	N	N	N	N	N
P	N	N	N	N	N	N	N	N

Проанализировав позиции, видим, что при правильной игре второго игрока первый всегда проигрывает. Несложно восстанавливается и сам стратегия: после каждого хода первого игрока, второй своим следующим ходом возвращает его на главную диагональ.

9.2 Игра Ним

Есть n кучек камней: $x_1, x_2, \dots, x_n (x_i > 0)$

Разрешается в свой ход взять любое количество камней, но только из одной кучки (в том числе все камни из кучки). Проигрывает тот, кто не может сделать ход.

$N = 1 : x_1$ - выигрывает *I* игрок (берёт всю кучку).

$N = 2 : x_1, x_2$ - можно посмотреть на эту ситуацию как на игру с ладьей, только можно ходить влево и вниз из верхнего правого угла (то есть за ход можно уменьшать одну координату, x_1 или x_2 . И тогда если $x_1 = x_2$ - выигрывает *II* игрок ровно так, как мы описывали выше, а если $x_1 \neq x_2$ - выигрывает *I*, первым ходом выравниваю количество камней в кучках и сводя задачу к предыдущей.

$N = 3$: позиция (n, n, m) , например, - *N*-позиция, т.к. она сводится к $(n, n, 0)$, *P*-позиции. Но в общем случае довольно сложно проанализировать позицию. Однако это возможно сделать с помощью двоичной СС.

Теорема: Каждую из n кучек записываем в двоичной системе счисления:

$$x_1 = x_1^{(r)} \dots x_1^{(1)} x_1^{(0)}$$

$$x_2 = x_2^{(r)} \dots x_2^{(1)} x_2^{(0)}$$

...

$$x_n = x_n^{(r)} \dots x_n^{(1)} x_n^{(0)}$$

Далее вычислим побитовый *XOR*, т.е. суммируем по модулю 2 биты у всех кучек в одном разряде. Получившуюся строку результатов будем называть *res*.

Наконец, теорема гласит, что (x_1, x_2, \dots, x_n) - *P*-позиция $\Leftrightarrow res = 00\dots0$

Пример:

Проанализируем позицию $(1, 2, 3)$:

$$1 = 01$$

$$2 = 10$$

$$3 = 11$$

$$res = 00 \Rightarrow \text{это } P\text{-позиция.}$$

Проанализируем позицию (2, 3, 4):

$$2 = 010$$

$$3 = 011$$

$$4 = 100$$

$res = 101 \Rightarrow$ это N -позиция.

Доказательство:

Докажем два факта (с их помощью можно будет применить анализ с конца, т.к. в заключительной позиции $res = 0$):

1. если $res \neq 0$, то можно сделать ход в позицию, где $res = 0$.
2. если $res = 0$, то любой ход ведёт в позицию, где $res \neq 0$.

1) Возьмём первую 1 слева в res . Пусть она соответствует некоторому k -му биту. Тогда должна найтись кучка x_i , у которой в k -м бите стоит 1. Заменяем эту единицу на 0. Далее идём вправо по двоичной записи x_i и инвертируем биты в тех столбцах, у которых в соответствующих позициях res стоит 1. Понятно, что получилось меньшее число x_i . То есть просто убираем из x_i -й кучи $x_i - x'_i$ камней и действительно получаем $res' = 0$. Этот факт доказан.

2) Этот факт совсем очевиден. Если мы уменьшаем кучку, мы точно поменяли какой-то его бит. Тогда понятно, что XOR по этой строке станет противоположным тому, что был, то есть 1. Значит $res \neq 0$. Теорема доказана.

9.3 Задача об угадывании числа

Есть числа $1, 2, \dots, N$. Есть два игрока: Алиса и Боб. Алиса загадывает число, а Боб может задавать вопросы, на которые можно ответить да или нет. Боб задаёт k вопросов. Его цель: точно определить загаданное число.

Одна из стратегий Боба заключается в том, что он может каждый раз выбирать среднее число, и спрашивать, находится ли загаданное число в первой половине (включая среднее). Если да, то он рассматривает первую половину (или примерно половину в случае нечётного). Иначе - вторую. Тогда понятно, что $k \leq \lceil \log_2 N \rceil$. Чтобы не было этой неудобной ситуации с тем, что нечетное число чисел не делится на две равные половины, можем предположить, что чисел изначально было, не N , а $2^s (2^s \geq N)$, где 2^s - наименьшая степень $2 \geq N$. Теперь Боб всегда может угадать число точно за s вопросов: $k \leq s = \lceil \log_2 N \rceil$.

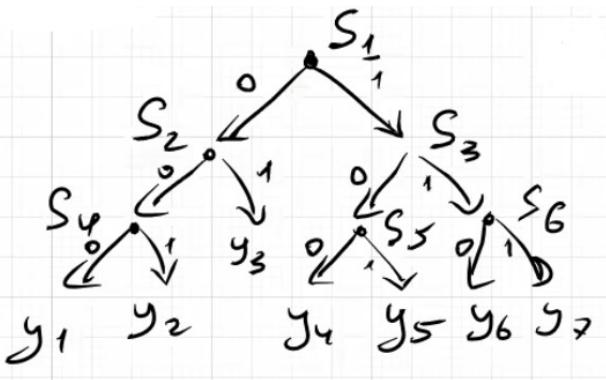
Поясним, почему не получится гарантированно угадать число за меньшее число вопросов k . Пусть хватило k вопросов: получили последовательность ответов $a_1 a_2 \dots a_k$ из да/нет, по которой мы восстанавливаем число. Понятно, что каждая последовательность однозначно задаёт своё число. Действительно, если для двух различных чисел x и y Алиса даёт Бобу на его вопросы полностью одинаковые ответы, то для Боба эти случаи неразличимы: его диалоги с Алисой для x и для y выглядят одинаково. При этом Боб после диалога выдаёт какой-то ответ, который определяется только состоявшимся диалогом. Значит в одном из случаев ответ будет неправильным. Далее, заметим, что не может быть так, что для двух различных x и y , загаданных Алисой, цепочка ответов для x является началом цепочки ответов для y . Действительно, иначе диалог Боба с Алисой выглядит одинаково для x и y до того момента, когда будут заданы все вопросы из цепочки ответов для x . Значит, к этому моменту Боб не может отличить x от y и должен делать для них одно и то же, тогда как он в одном случае задаёт следующий вопрос, а в другом - нет.

Таким образом, мы получили, что каждому числу от 1 до N соответствует последовательность из не более чем k нулей и единиц, все эти последовательности различны, и ни одна не является началом другой. Заметим, что семейство этих последовательностей содержит не более 2^k элементов. Действительно, если какая-то из них имеет длину меньше k , то продолжим её, например, нулями. Тогда для различных x и y полученные последовательности длины k различны: иначе они либо совпадают, либо одна (более короткая) является началом другой. Таким образом, каждому числу от 1 до N соответствует последовательность длины k из нулей и единиц, и все эти последовательности различны. Всего последовательностей длины k из нулей и единиц 2^k . По принципу Дирихле, чисел от 1 до N должно быть не больше 2^k (иначе двум разным числам соответствуют одинаковые последовательности). Значит нужно, чтобы $2^k \geq N \Rightarrow k \geq \lceil \log_2 N \rceil$ ■

9.4 Разрешающие деревья

Даны X, Y - конечные множества и $f : X \rightarrow Y$. Требуется вычислить $f(x)$ при некотором неизвестном входе x . Разрешается задавать вопросы типа $x \in S$ для подмножеств S множества X . Тогда разрешающим деревом будем называть двоичное дерево, каждая промежуточная вершина которого (не лист) помечена некоторым подмножеством $S \subseteq X$. Каждый лист помечен элементом $b \in Y$. Из каждой промежуточной вершин выходит три ребра: одно к корню и два к листьям. Для каждой промежуточной вершины одно из рёбер, ведущих к листьям, помечено единицей, а другое - нулюм.

Некоторый конкретный пример разрешающего дерева:



Мы строим путь от корня на каждом шаге переходя по ребру с 1, если ответ положительный, и 0, если ответ отрицательный. Вычисления завершается тогда, когда мы попадаем в лист. Мы говорим, что протокол (по-другому разрешающее дерево) вычисляет функцию f , если для всякого $x \in X$ протокол выдаёт $f(x)$.

Сложностью адаптивного протокола (адаптивным протоколом - такой протокол, вопрос в котором может зависеть от предыдущих ответов) называется глубина дерева (несложно убедиться, что она равна количеству вопросов, которое потребуется задать в худшем случае).

Рассмотрим неадаптивную постановку задачи. Предположим, что Боб отправляет сразу k вопросов по почте. Тогда можем задавать такие независимые вопросы: верно ли, что в i -м бите числа стоит 1? $\forall i$. Битов в числе не более чем $\lceil \log_2 N \rceil$, значит $k \geq \lceil \log_2 N \rceil$. Понятно, что меньше вопросов, чем в адаптивном точно не получится, поскольку неадаптивный протокол не опирается на историю ответов, что несколько усложняет вычисление (ведь если Боб в неадаптивном протоколе справляется с задачей за k вопросов, то он точно справится за k и в адаптивном - достаточно задать те же вопросы).

Давайте покажем на изображённом выше примере (к задаче об угадывании числа), как переписать адаптивный протокол в неадаптивный. Приведём вопросы, которые мы будем задавать:

- 1) Верно ли, что $x \in S_1$?
- 2) Верно ли, что если $x \in S_1$, то $x \in S_3$, а если же $x \notin S_1$, то $x \in S_2$?
- 3) ...

Вопросы будут получаться большими и громоздкими, однако данное рассуждение работает для любой задачи такого типа и любого адаптивного алгоритма.

9.5 Примеры задач

9.5.1 Задача о взвешивании

Есть n монет различной массы, ход заключается в сравнении весов двух монет. Наша цель: найти самую тяжёлую монету за как можно меньшее число вопросов k .

Решение:

Лемма. Для адаптивной модели необходимо и достаточно $n - 1$ взвешивание.

Доказательство:

Достаточность: можно взять любую монету и сравнивать её с какой-нибудь другой. Если она оказалась тяжелее, то выкидываем ту, с которой сравнивали и продолжаем сравнение с оставшимися. Если же она оказалась легче, то выкидываем её и берём ту, которой она "проиграла" и уже с ней продолжим сравнение с оставшимися. Таким образом за $n - 1$ сравнение мы вычислим максимум.

Необходимость: пусть мы нашли максимум за $n - 2$ взвешивания. Тогда построим граф, где вершинами будут монеты, а ребрами будут соединены те монеты, которые мы попарно сравнивали. Если в графе на n вершинах $n - 2$ ребра, то он несвязен. Тогда рассмотрим его компоненту связности V_1 , в которой находится максимум, и V_2 - все остальные объекты. Увеличим веса всех монет в V_2 на одно и то же число так, чтобы они были тяжелее максимума в V_1 . Причём результаты взвешиваний не изменятся, потому что они были или в пределах V_1 , или в пределах V_2 , а самая тяжёлая монета станет другой (теперь она в V_2). Таким образом результат взвешивания даст один и тот же результат при разных максимумах, значит алгоритм работает неправильно. Противоречие, значит $n - 1$ взвешиваний необходимо.

Теперь посмотрим на неадаптивную модель для данной задачи. Как мы видели ранее в общей модели разрешающих деревьев, отличий между адаптивным и неадаптивным протоколом нет, их сложности одинаковые. Но это достигается за счёт весьма нетривиальных запросов в неадаптивной модели. Если же мы ставим ограничения на тип запросов, то логично ожидать, что разница между моделями появится.

Лемма. Докажем, что для неадаптивной модели необходимо и достаточно $\binom{n}{2} = \frac{n(n-1)}{2}$ взвешивания.

Доказательство:

Достаточность: очевидна, т.к. можем просто сравнить попарно все объекты и найти максимум. Как раз будет $\binom{n}{2}$ сравнения.

Необходимость: пусть мы нашли максимум меньше чем за $\binom{n}{2}$ взвешивания. Значит есть два объекта x и y , которые не сравнивались. Тогда подадим на вход те же объекты, но увеличим массу у x . А затем еще один вход с теми же объектами, но увеличим массу у y . Мы получим одинаковые ответы при разных входах, а значит алгоритм выдаст на них один и тот же результат. Поскольку самые тяжёлые объекты в этих двух входах разные, алгоритм работает неправильно. Противоречие, значит $\binom{n}{2}$ сравнения необходимо.

9.5.2 Задача о связности графа

Для начала определим, что такая сложность протокола для вычисления булевой функции: нам дана $f : \{0, 1\}^n \rightarrow \{0, 1\}$, причём сам набор x_1, \dots, x_n не дан. Мы можем задавать вопросы вида: верно ли, что $x_i = 0$. Однако есть функции, для которых можно задать меньше вопросов, чтобы вычислить её значение. Тогда сложность протокола - минимальное число вопросов указанного вида, которое нужно задать, чтобы вычислить $f(x_1, \dots, x_n)$.

В доказательствах нижних оценок используется метод противника. Будем играть в модели разрешающих деревьев. Противник задаёт какой-нибудь вход в алгоритм, в процессе возможно видоизменяя его, но так, чтобы его ответы согласовывались с предыдущими его ответами и не противоречили реальности. Тогда алгоритм пытается выдать ответ за как можно меньшее число вопросов, а противник (за него будем играть мы) наоборот пытается увеличить это число вопросов. То есть алгоритм играет за MIN, а мы играем за MAX. У такой игры есть цена k , то есть такое число вопросов, что алгоритм всегда может выдать правильный ответ за $\leq k$ вопросов, а противник задать такой вход, чтобы алгоритму нужно было задать $\geq k$ вопросов. Тогда число k - это будет необходимое число вопросов.

Решим такую задачу: нужно проверить связность графа. Введём булеву функцию $CONN : \{0, 1\}^{\binom{n}{2}} \rightarrow \{0, 1\}$ от $\binom{n}{2}$ переменных, где x_{ij} - есть ребро ли ребро (1), или нет (0). То есть функция $CONN$ принимает граф (множество его рёбер) и выдаёт 1, если он связан, 0 - в противоположном случае.

Нужно узнать сложность булевой функции в модели, указанной выше.

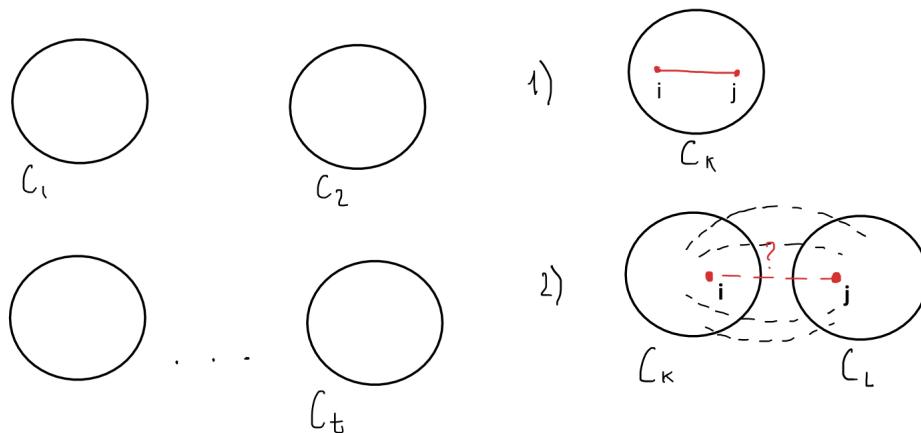
Утверждение: Сложность функции $CONN$ в модели булевых функций = $\binom{n}{2}$

Доказательство:

Достаточность: Достаточность очевидна, можно просто задать вопрос про каждое ребро (которых $\binom{n}{2}$ штук) и в получившемся графе проверить связность.

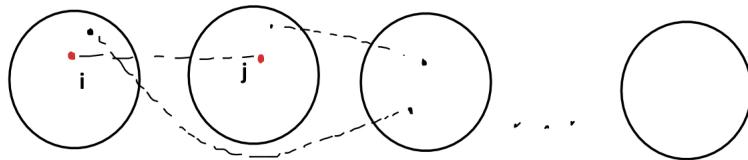
Необходимость: Играем за противника. В данной игре на самом деле даже не важна согласованность действий противника, поскольку рёбра не зависят друг от друга (хотя, конечно, если вопрос повторяется дважды - каждый раз нужно отвечать одно и то же). Так что требуется описать только стратегию противнику.

На конкретном ходе у нас есть какие-то компоненты связности (изначально точки). Есть несколько ситуаций, в зависимости от того, про какие рёбра был задан вопрос:



1. Вопрос задан про ребро между вершинами i и j внутри одной компоненты связности. В таком случае $x_{ij} = 1$ (если эта переменная не спрашивалась ранее).
2. Вопрос задан про ребро между вершинами i и j из разных компонент связности. В таком, случае, если про все рёбра из C_k в C_l были даны ответы "нет"(0), то $x_{ij} = 1$, иначе $x_{ij} = 0$.

Предположим, что алгоритм не спрашивал про некоторое ребро x_{ij} и при этом выдал ответ. Чтобы прийти к противоречию, нужно найти два входа и со связным и не связным графом, в зависимости от того, какое ребро находится на месте x_{ij} .



Посмотрим на то, есть ли путь между вершинами i и j (это и нужно для связности). Во-первых, заметим, что при расширении компоненты связности (когда мы соединяем две), все ребра внутри компоненты связности нам известны (показывается индуктивно). Поэтому i и j не могли быть связаны путём (иначе они лежат в одной компоненте связности, но все рёбра внутри компоненты связности должны быть известны, а про x_{ij} информации нет). Значит i и j лежат в разных компонентах связности. Могут быть и другие компоненты, не содержащие i и j , но при этом не по построению не бывает, что между двумя компонентами рёбер нет совсем (тоже сохраняется индуктивно). Значит алгоритм просто не спросил про нужное ребро между этими компонентами. Поэтому если всех этих неспрошенных рёбер нет, то граф несвязен. А если они все есть - то он связан. Поэтому нельзя не спросить про какое-то ребро, иначе мы точно не можем определить, связан график или нет. Значит $\binom{n}{2}$ вопросов необходимо. Понятно, что и в неадаптивном протоколе меньшее число вопросов нельзя было задать (неадаптивный только сложнее).

10 Лекция 22

(Андрей)

10.1 Булевы схемы

Вначале введем понятие булевых функций (отображений).

Булевые функции - это такие функции f , что они определены на множестве $\{0, 1\}^n$ и принимают значения в множестве $\{0, 1\}^k$. То есть из n переменных мы получаем k -элементный вектор из 0 или 1.

Булева схема - последовательность булевых функций $g_1, g_2, \dots, g_s : \{0, 1\}^n \rightarrow \{0, 1\}$, где:

$$g_i = \begin{cases} h_j \cup h_k \\ h_j \cap h_k \\ \neg h_j \end{cases}$$

В свою очередь, $h_j = \begin{cases} x_r, r = \{1, \dots, n\} \\ g_m, m < i \end{cases}$

Случай $h_j = g_m, m < i$ означает, что мы можем пользоваться тем, что мы сконструировали ранее.

Размер схемы - количество функций g_i .

Сложность схемы - наименьший размер схемы, вычисляющей данную функцию.

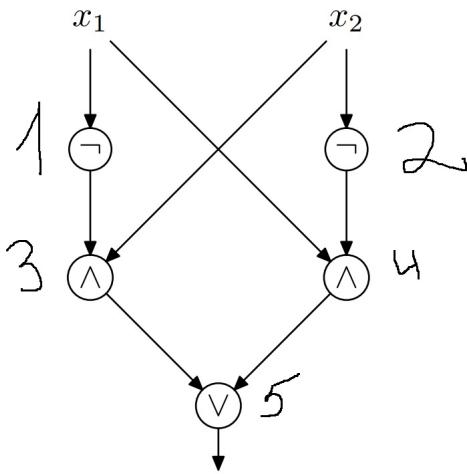
Говорят, что схема вычисляет функцию f , если $\forall x \in \{0, 1\}^n f(x) = (g_{s-k+1}(x), g_{s-k+2}(x), \dots, g_s(x))$ Обратим внимание, что т.к. исходная функция выдает ровно k чисел, то и на выходе мы получили именно k элементов нашей схемы, причем это будут последние k элементов. (ни один филолог не пострадал при прочтении этого предложения).

Также эти функции $(g_{s-k+1}(x), g_{s-k+2}(x), \dots, g_s(x))$ называют выходами функции.

Пример: x_1, x_2

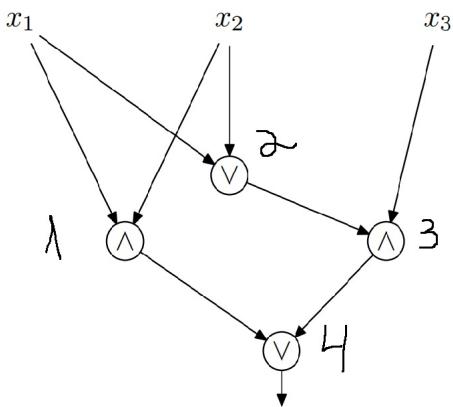
Проделаем следующие операции: $\neg x_1, \neg x_2, \neg x_1 \wedge x_2, \neg x_2 \wedge x_1, (\neg x_1 \wedge x_2) \vee (\neg x_2 \wedge x_1)$. Такая схема с одним выходом вычисляет $x_1 \oplus x_2$.

Иногда бывает полезно визуализировать булевые схемы в виде ориентированного графа:



Причем стоит отметить, что такой граф всегда будет ациклическим в силу топологической сортировки.

Приведем такую же визуализацию для функции MAJ_3 :



10.2 Сложение двоичных чисел

Пусть нам даны две n -битовых двоичных записи чисел x и y и мы хотим вычислить двоичную запись их суммы $z = x + y$. Для удобства обозначим $x = x_{n-1} \dots x_1 x_0$, где x_0 младший разряд двоичной записи. Аналогично, $y = y_{n-1} \dots y_1 y_0$. Во-первых, заметим, что в двоичной записи z будет не более $n + 1$ разрядов. Так что мы хотим построить схему с $2n$ входами и $n + 1$ выходом.

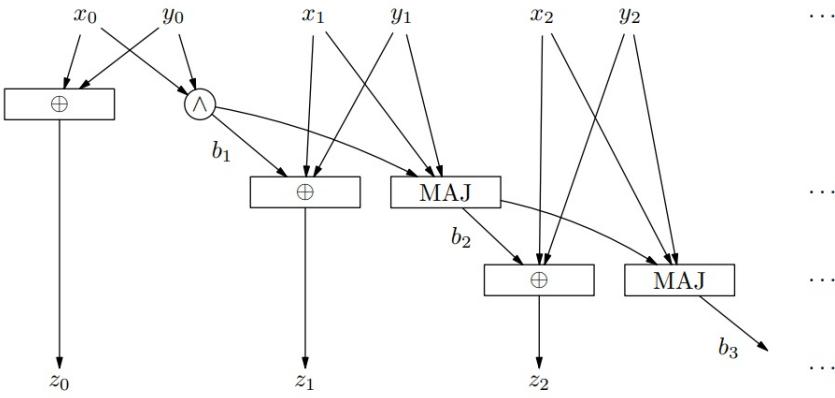
Идея конструкции схемы будет та же, что и в обычном школьном сложении в столбик. Мы будем складывать числа x и y поразрядно, попутно вычисляя биты переноса в следующий разряд.

Для удобства будем обозначать через b_i бит, который переносится в i -ый разряд из предыдущих.

Заметим, что мы уже готовы вычислить первый разряд ответа $z_0 = x_0 \oplus y_0$. Конечно, мы не можем сразу применить операцию \oplus , но выше мы показали, как её можно вычислить небольшой схемой. Добавим эту маленькую схему в нашу как подсхему. Далее, заметим, что $b_1 = x_0 \wedge y_0$, добавим соответствующий элемент в схему. Переидём к следующему разряду. Здесь $z_1 = x_1 \oplus y_1 \oplus b_1$ и $b_2 = MAJ_3(x_1, y_1, b_1)$.

Для вычисления первого выражения добавим сначала подсхему, вычисляющую промежуточную величину $c_1 = x_1 \oplus y_1$, а затем подсхему, вычисляющую $z_1 = c_1 \oplus b_1$. Для вычисления b_2 просто добавим подсхему, вычисляющую функцию MAJ_3 . Такая схема также приведена выше. Дальше, случай произвольных z_i и b_i полностью аналогичен случаю z_1 и b_1 и мы можем последовательно вычислить все эти значения.

Приведем пример визуализации для сложения чисел:



Оценим теперь размер описанной схемы. Для каждого разряда ответа нам нужно не больше двух раз применить подсхему для вычисления функции \oplus и не более одного раза подсхему для вычисления MAJ_3 . Все эти схемы имеют фиксированный размер, так что для вычисления каждого разряда z мы используем фиксированное число элементов, не зависящее от числа входных переменных. Поэтому всего в схеме $O(n)$ элементов.

10.3 Вычисление произвольной функции

Теорема: Пусть имеется функция $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Тогда такая функция может быть вычислена схемой размера $O(n \cdot 2^n)$.

Доказательство:

Мы знаем, что любая булева функция представляется в виде СДНФ:

$$f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{\alpha_1, \alpha_2, \dots, \alpha_n \\ f(\alpha)=1}} x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$$

Напомним: $x^\alpha = \begin{cases} x, & \alpha = 1 \\ \neg x, & \alpha = 0 \end{cases}$

Понятно, что число дизъюнкций не превышает 2^n , а число элементов в каждом из конъюнктов не превышает n . Таким образом, при вычислении функции на все дизъюнкции мы потратим не более 2^n операций, а внутри каждого конъюнкта мы используем не более $n + n - 1$ операции (может быть n отрицаний, и еще надо добавить $n - 1$ конъюнкцию). Итого получили, что общее число операций не превысит $2^n + 2^n(2n - 1) = 2n \cdot 2^n = O(n \cdot 2^n)$ операций.

Теорема: Для всякого $n > 10$ существует функция $f : \{0, 1\}^n \rightarrow \{0, 1\}$, которую нельзя вычислить схемой размера меньше $\frac{2^n}{10n}$.

Доказательство:

Для доказательства применим мощностной метод: докажем, что функций больше, чем схем нужного размера. Тогда схем не хватит, чтобы вычислить все функции.

Понятно, что булевых функций от n переменных 2^{2^n} .

Оценим количество схем размера S (в нашем случае $S = \frac{2^n}{10n}$). Причем стоит отметить, что если мы вычислили функцию схемой размера меньше S , то мы можем добавить некоторое количество фиктивных переменных так, чтобы размер схемы стал равен S .

Посчитаем количество бит, необходимых для задания схемы. Теперь сделаем для себя некоторое удобство, чтобы было легче описать нашу схему, — а именно, выпишем все элементы схемы, включая переменные:

$$x_1, x_2, \dots, x_n, g_1, g_2, \dots, g_s$$

Теперь надо понять, как мы можем закодировать нашу последовательность. Для каждого элемента нам нужно сделать две вещи:

1. Указать два предыдущих элемента (в смысле из каких двух элементов был составлен данный).
2. Указать, какая операция была использована, чтобы получить данный элемент (конъюнкция, дизъюнкция или отрицание).

Для первого случая нам потребуется не более $2 \cdot \log_2(n + S)$ бит (для каждого из двух предыдущих элементов потребуется не более $\log_2(n + S)$ бит).

Далее, для второго случая нам достаточно 2 бит, чтобы указать какой тип операции был использован.

Итого для каждого из S элементов в схеме потребуется $2(1 + \log_2(n + S))$ бит. Тогда для всей схемы получаем итоговый размер $2S(1 + \log_2(n + S))$ бит. Сделаем оценку на полученное число:

$$2S(1 + \log_2(n + S)) \leq 2S \cdot (2 + \log_2 S) \leq 4S \cdot \log_2 S$$

При $S = \frac{2^n}{10n}$ получаем:

$$4 \cdot \frac{2^n}{10n} \cdot (n - \log_2(10n)) \leq 2 \cdot \frac{2^n}{5}$$

Поэтому количество схем размера S не больше, чем количество таких строк, то есть не больше $2^{2 \cdot \frac{2^n}{5}}$, что меньше, чем кол-во всех булевых функций (2^{2^n}), следовательно, не всякую функцию можно вычислить схемой размера S .

Ч.Т.Д

11 Лекция 23

(Илья)

11.1 Функция XOR_n

Мы знаем, что $XOR_n = x_1 \oplus x_2 \oplus \dots \oplus x_n$

Определение: Схемная сложность функции - минимальное количество элементов в схеме, вычисляющей данную функцию.

Тогда давайте сформулируем теорему:

Теорема: Схемная сложность $XOR_n \geq 2n - 1$

Доказательство:

- Доказательство будем вести по индукцией по n , однако хитрой индукцией. Сразу для функции XOR_n и её отрицания - $\neg XOR_n$

Мы раньше говорили, что в схеме будем допускать только элементы \vee, \wedge, \neg , однако в этом доказательстве будем допускать схемы более общего характера, поэтому будем допускать, что элементами схемы могут быть: $(h_i^a \wedge h_j^b)^c$

Тут имеется в виду, что $x^a = \begin{cases} x, & \text{если } a = 1 \\ \neg x, & \text{если } a = 0 \end{cases}$

Что такое h_i и h_j ?

Ну вот раньше у нас были функции g_1, g_2, \dots, g_s , такие что $g_k = \begin{bmatrix} \neg h_i \\ h_i \wedge h_j \\ h_i \vee h_i \end{bmatrix}$, где h_p - это или предыдущий элемент или переменная, которая встречалась раньше.

Теперь же мы просто рассматриваем более общие элементы.

Примеры:

- $a = b = c = 1$, тогда получаем $h_i \wedge h_j$
- $a = b = c = 0$, тогда получаем $h_i \vee h_j$
- $i = j$ и $a = b = 1$, тогда получим или h_i , или $\neg h_i$ в зависимости от того, чему равно c
- $i = j$ и $a = 0, b = 1$, тогда мы получаем или 0, или 1

Отсюда видно, что размер новой схемы будет точно не меньше размера новой схемы, потому что элементы старой схемы включены в элементы новой схемы, что мы и показали в примерах, однако в новой схеме мы можем получать и другие элементы другими комбинациями a, b, c .

- Теперь запускаем индукцию.

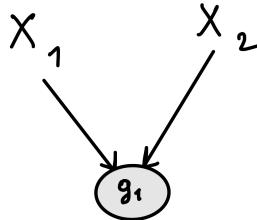
База: $n = 1$ и получим, что нам будет необходимо для $XOR_1 = x_1$ минимум 1 элемента в схеме. Ну это правда, потому что нам нужен какой-то выход в схеме (тут, конечно, вопрос с обозначениями, но мы выход из схемы в данном случае считаем за элемент).

Переход: $n - 1 \rightarrow n$

Рассмотрим "новую" схему минимального размера, которая имеет строго меньше, чем $2n - 1$ элемента, которая вычисляет XOR_n или $\neg XOR_n$.

У нас есть n переменных x_1, x_2, \dots, x_n и вот в этой схеме наши элементы как-то устроены, пусть это $g_1, g_2, \dots, g_s = (XOR_n)^\alpha$

Берём первый элемент g_1 , понятно, что в первый элемент мы подставляли две переменные или одну переменную. Не умоляя общности давайте обозначим эти переменные за x_1 и x_2 , тогда имеем, что



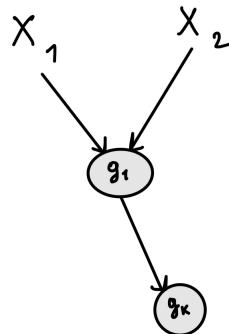
Замечание: $(x_i^a \wedge x_j^b)^c \neq x_i \oplus x_j$ ни при каких a, b, c , потому что \oplus принимает две единицы и два нуля, а та конъюнкция истинна только на одном наборе (x_i, x_j)

Это замечание просто к тому, чтобы было понятно, что мы не сможем на первом шаге вычислить XOR_2 , то есть $g_1 \neq XOR_2$.

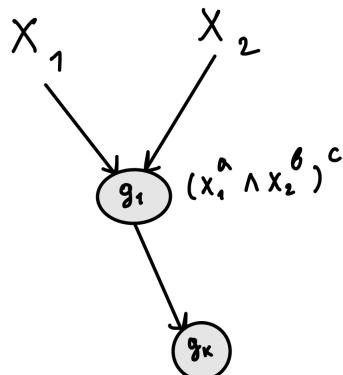
Значит, элемент g_1 участвовал в дальнейшей реализации нашей схемы, то есть он поступал в какой-то элемент g_k . (может быть в несколько поступал, но вот мы конкретное место выберем и посмотрим туда)

Почему? Ну если это не так, то он не помогал вычислять XOR_n , а значит бы не нужный этой схеме, поэтому мы бы его выкинули и получили бы схему меньшего размера, но по условию схема была минимального размера.

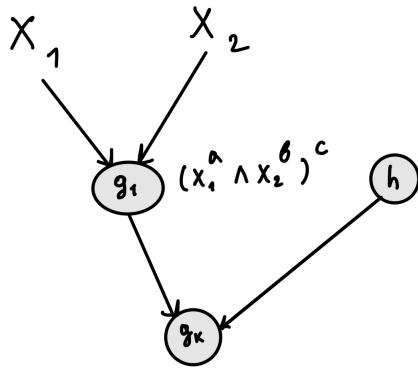
Значит, имеем вот, что



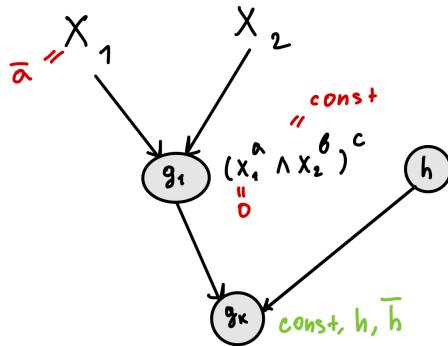
Давайте подпишем, что такое g_1 .



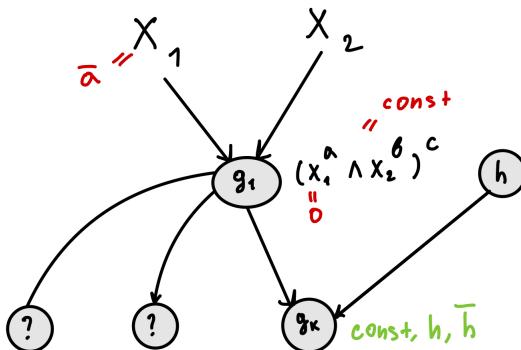
А g_k - это тоже элемент которому на вход что-то приходило, одна из частей входа - это g_1 , а вторая пусть будет h .



Давайте теперь поменяем вход и подставим вместо $x_1 - \neg a$ и получим, что $g_1 = ((\neg a)^a \wedge x_2^b)^c = (0 \wedge x_2^b)^c = 0^c$. То есть мы получили, что g_1 стал константой. Тогда что можно сказать про g_k ? Ну на самом деле так как там стоит $(g_1^a \wedge h^b)^c$, а g_1 - это или 0 или 1, то получаем, что в g_k может стоять или константа, или $\neg h$, или h



Ну от g_1 вероятно мог подаваться не только на g_k , но и в какие-то другие элементы, изобразим их.



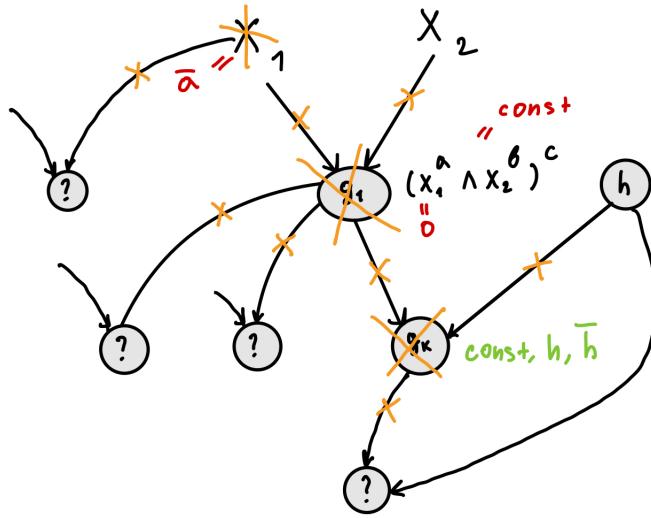
Теперь мы делаем следующее. Мы удаляем из схемы g_1, g_k, x_1 .

Что произошло с функцией после всех предыдущих манипуляций? Ну после подстановки в XOR_n вместо одной из переменных константы мы получаем или XOR_{n-1} или $\neg XOR_{n-1}$. Почему? Ну просто из-за того, что $1 \oplus x_1 \oplus x_2 = \neg(x_1 \oplus x_2)$ и $0 \oplus x_1 \oplus x_2 = x_1 \oplus x_2$

Однако теперь схема может стать не совсем корректной, поэтому нам надо поправить график, который получился, чтобы он стал настоящей схемой

Какая у нас есть проблема? У нас g_k, g_1, x_1 кому-то могли подаваться на вход.

Но на самом деле проблемы нет, потому что мы можем просто заметить, что так как x_1 и g_1 стали константами, то мы можем, если они поступали куда-то, просто оставить второй вход, который шёл к этим элементам. А что касается g_k , то так как он что-то из $const, h, \neg h$, то можем направить, если он куда-то поступал, в его входной элемент просто h , потому что за счёт второго входа мы сможем использовать как раз $const, h, \neg h$.



Ну тут есть ещё небольшой нюансик с тем, что, кто нам сказал, что h - это функция, ведь это может быть переменной. Но на самом деле это не проблема, а вот что если у нас h был равен g_1 или x_1 , например? Ну тогда раз это у нас константы, то по-прежнему всё хорошо, а если он был равен g_k , то у же было сказано, что делать - надо заменять просто на h .

Итого, мы получили корректную схему, вычисляющую XOR_{n-1} или $\neg XOR_{n-1}$ размера $< 2n-3$, но по предположению индукции у нас схема была размера $\geq 2n-3$ - противоречие!

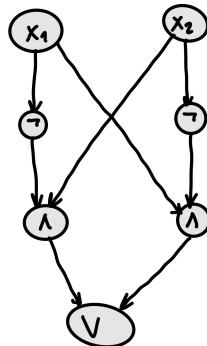
Значит схемная сложность $XOR_n \geq 2n-1$

Ч.Т.Д.

То, что мы сейчас доказали - это доказали нижнюю оценку.

Однако очень просто можно показать верхнюю оценку.

Заметим, что для выражения XOR_2 нам понадобится 5 операций



А для XOR_3 нам понадобится ещё 5 операций, ну и тд, поэтому понятно, что верхняя оценка на сложность схемы XOR_n - это $5(n-1)$

11.2 Глубина схемы

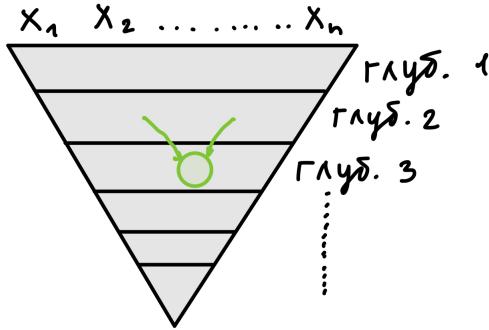
Определение: Глубина элемента g_i в нашей схеме - это длина наибольшего пути от переменной к g_i .

Глубина всей схемы - это максимальная из глубин элементов.

Например, глубина схемы для XOR_2 , приведённой чуть выше равна 3.

Теперь представьте себе, что у вас есть булева схема и вы можете разбить элементы по глубине, как бы по уровням.

Ну и в элемент, скажем, глубины 3 могут вести стрелки только из элементов меньшей глубины. Ну иначе у него глубина была бы больше, чем 3.



Пример:

- Давайте вычислим глубину дизъюнкции. $x_1 \vee x_2 \vee \dots \vee x_n$

Это можно сделать последовательно и получить глубину $O(n)$

А можно сделать не так и разбить их на пары и получить меньшую длину, то если как бы по двоичному дереву. Получим глубину $O(\log_2(n))$

- Давайте посчитаем пример, связанный с функцией $CONN : \{0,1\}^{\binom{n}{2}} \rightarrow \{0,1\}$

Хочется, понять, как можно просто построить схему, которая её вычисляет и при этом может быть сэкономить на глубине и понять какой тут размер в теории возможен.

Мы вводим, так называемую модифицированную матрицу смежности:

Мы берём граф G и рассматриваем матрицу $A(G)$ - это матрица n на n , состоит из элементов

$$a_{ij} = \begin{cases} 1, & \text{если вершины } v_i \text{ и } v_j \text{ соединены ребром и } i \neq j \\ 0, & \text{если вершины } v_i \text{ и } v_j \text{ не соединены ребром и } i \neq j \\ 1, & \text{если } i = j \end{cases}$$

У матрицы смежности по диагонали обычно стоят нули, но мы поставим единицы, потому что единицы будут соответствовать петлям. И мы понимаем, чтобы проверить, связан ли граф, надо просто понять, есть ли путь просто вершины u в v какой-то длины и длину эту можно ограничить числом вершин, то есть длина должна быть не больше, чем $n - 1$. Но за счёт того, что у нас есть петли мы сможем найти путь длины ровно $n - 1$. И вот эта информация как раз и хранится в степенях матрицы $A(G)$

Утверждение: Матрица A^k на позиции a_{ij} содержит количество путей длины k из v_i в v_j .

Доказательство:

1. Будем вести индукцию по k .

База: $k = 1$ - очевидно.

Переход: $k - 1 \rightarrow k$.

$$A^k = A^{k-1} \cdot A, \text{ а } a_{ij}^{(k)} = \sum_{t=1}^n a_{it}^{(k-1)} \cdot a_{tj}$$

Однако посмотрим на то, что написано с комбинаторной точки зрения. Вот у нас есть вершины v_i, v_t, v_j и мы хотим посчитать количество путей длины k из v_i в v_j . Для этого мы проведём одно ребро, если оно есть из v_j в v_t и будем считать количество путей длины $k - 1$ из v_i в v_t . Как раз тогда a_{tj} будет отвечать за наличие ребра между v_t и v_j и тогда мы просто суммируем по всем таким вершинам v_t , а это ровно то, что у нас и написано.

Ч.Т.Д.

Ну и при построении схемы возникает потребность в подсчёте A^{n-1} . Чтобы это сделать, давайте будем возводить матрицу в степень $n - 1$ в булевом смысле, то есть сложение заменим на \vee , а умножение на \wedge .

Тогда конкретно формула матричного произведения принимает вид: $(A \cdot B)_{(ij)} = \bigvee_{t=1}^n A_{(i,t)} \wedge B_{(t,j)}$

Однако теперь,

$$A_{(i,k)}^k = \begin{cases} 1, & \text{если } \exists \text{ путь длины } k \text{ из } v_i \text{ в } v_j \\ \text{иначе} & \end{cases}$$

Поэтому теперь нам достаточно возвести матрицу в $n - 1$ степень и проверить, что она состоит из всех единиц.

Ну на самом деле удобно вычислять степень не $n - 1$, а $2^m (2^m \geq n) \Rightarrow m = \lceil \log_2(n) \rceil$)

какую-то, для быстрого возвведения в степень.

Ну и тогда получаем, что на каждом шаге мы будем возводить матрицу в квадрат. Это занимает порядка n^3 действий, ну и таких шагов будет m штук, поэтому схема будет размера порядка $O(n^3 \log_2(n))$ Ну там ещё надо будет взять конъюнкцию по всем n^2 элементам матрицы, чтобы проверить, что там все единички, ну это ещё n^2 действий, но сложность это не поменяет.

А вот , что касается глубины, то мы делаем m операций и каждую матрицу мы вычисляем дизьюнкцией и глубина там $\log_2(n)$ (было обсуждено в первом примере), поэтому глубина здесь $O(\log_2^2(n))$, ну на самом деле там ещё нужно из-за конъюнкции $\log_2(n^2)$, итого выйдет $O(\log_2^2(n) + \log_2(n^2)) = O(\log_2^2(n))$.

Здесь можно немножечко сэкономить в размере и посчитать не за $n^3 \log_2(n)$, а за n^3 , потому что на самом деле нам не нужна информация про всю матрицу, а нужна информация только про её первый столбец, потому что если из первой вершины достижимы все другие, то это уже означает, что граф связан.

То есть должно быть выполнено вот такое:

$$A^{n(\geq n-1)} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

•

Умножение здесь подразумевается булево.

Теперь мы можем вычислять с конца:

$$\left(A \dots \left(A \left(A \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right) \right) \dots \right)$$

Теперь умножение матрицы на столбец занимает n^2 операций, ну и делаем так n раз и получим порядка n^3 операций. То есть размер - это $O(n^3)$

А глубиной мы теперь пожертвовали и теперь она равна $O(n \log_2(n))$

11.3 Формулы

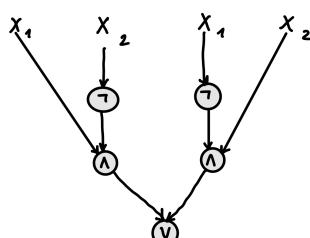
- x_i - это формула
- Φ - формула, то и $\neg\Phi$ тоже
- Φ_1, Φ_2 - формулы, то и $\Phi_1 \wedge \Phi_2$ и $\Phi_1 \vee \Phi_2$ тоже формулы

Ну и теперь за конечное число шагов можно получить любую формулу.

Пример:

Формула для $XOR_2 = \neg x_1 \wedge x_2 \vee \neg x_2 \wedge x_1$

Формулу тоже можно рисовать как график, тогда получим, что



Чем этот граф отличается от булевой схемы?

Ну на самом деле исходящие степени всех вершин равны 1, ведь в булевой схеме это может быть не так, даже на примере XOR_2 (чуть выше была картинка).

Поэтому формула - это частный случай булевой схемы, она тоже имеет глубину и размер. Между ними имеется связь и оказывается, что любую схему можно переделать в формулу.

Утверждение: Для любой схемы C от переменных x_1, x_2, \dots, x_n размера s и глубины d существует эквивалентная ей формула размера не больше, чем $2^s - 1$ и глубины не больше, чем d .

Доказательство:

1. Доказательство можно вести индукцией по d и по s одновременно, можно и по их сумме, но возьмём на примере s .

Итого, индукция по s .

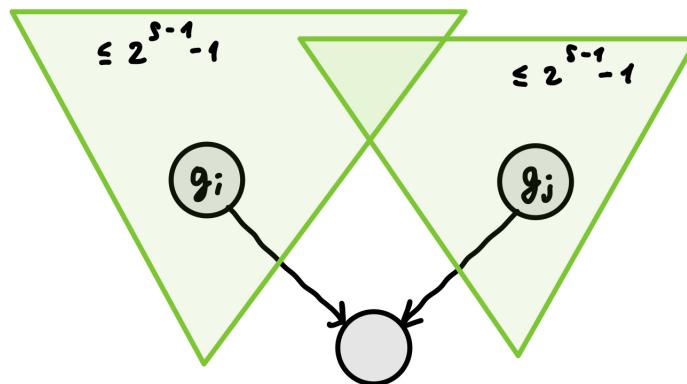
База: $s = 1$ - очевидно.

Переход: $s - 1 \rightarrow s$.

Рассмотрим теперь промежуточный случай, когда два элемента подаются на вход другому элементу.

То есть пусть на вход подаются два элемента - g_i и g_j .

Ну и рассмотрим схемы, которые были ранее.



Ну и тогда суммарный размер формулы не превосходит $(2^{s-1} - 1) \cdot 2 + 1 = 2^s - 1$

Аналогично, с глубиной, она не превышала у g_i и g_j $d-1$, тогда суммарная глубина не превышала $d - 1 + 1 = d$.

Ч.Т.Д.

Следствие: Схема глубины d допускает эквивалентную реализацию тоже глубины d и размера не больше, чем 2^d .

Доказательство:

1. Переделываем её в формулу и получаем дерево, не обязательно полным двоичным деревом. Поэтому глубина у нашего дерева d , а листьев(то есть операций) у этого дерева не больше, чем 2^d .

Ч.Т.Д.

11.4 Задачи выполнимости

Их бывает несколько вариаций:

1. Для схем.

Нам даётся схема C на вход и спрашивают, существует ли x , такой, что $C(x) = 1$, $x = (x_1, x_2, \dots, x_n)$

2. КНФ - конъюнкция элементарных дизъюнктов.

Есть некая КНФ K и спрашивают, существует ли такой x , что $K(x) = 1$.

3. 3-КНФ - частный случай КНФ, где каждый дизъюнкт имеет не более трёх переменных.

Например, $(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_2) \wedge (\neg x_5)$

Почему эти задачи так важны? Ну на самом деле, если мы умеем отвечать на вопрос про КНФ-3, то тогда мы сможем эффективно свести к задаче КНФ. Эффективно значит свести полиномиально. Аналогично с вопросами про схему.

12 Лекция 24

(Никита)

12.1 Задача выполнимости

Давайте напомним, какие типы задач выполнимости мы рассматриваем:

- Схема $C(x)$

$$\exists x \in \{0, 1\}^n : C(x) = 1?$$

- КНФ $K(x)$

$\exists x \in \{0, 1\}^n : K(x) = 1?$ Напомним, что $K(x) = D_1 \wedge D_2 \wedge \dots \wedge D_r$, где $D = x_1^{\alpha_1} \vee x_2^{\alpha_2} \vee \dots \vee x_s^{\alpha_s}$. Факт для расширения развития терминологии: $x, \neg x$ называют *литералами*.

- 3-КНФ: КНФ, в которой каждый D_j содержит ≤ 3 литералов.

$$\exists x \in \{0, 1\}^n : K(x) = 1?$$

Отметим, что каждая функция имеет КНФ (подумайте, это напрямую следует из существования СДНФ и закона Де Моргана).

Мы хотим подавать некоторой программе (solver для 3-КНФ) некоторый вход и хотим получить ответ на задачу выполнимости. На вход подаём нашу задачу и хотим, чтобы за полиномиальное время (например, от размера схемы, которая подаётся) наша задача сводилась к 3-КНФ.

Упражнение. Не любая схема обладает 3-КНФ (подумайте, почему!). Один из таких примеров: XOR некоторого большого числа переменных.

Что же тогда делать для таких функций? Для них мы построим некоторую другую функцию (например, добавляя переменные) так, чтобы она была выполнима тогда и только тогда, когда выполнима исходная.

Свести КНФ к схеме несложно - просто пошагово построить схему. 3-КНФ вообще к КНФ сводить не нужно - это её частный случай. Трудность вызывает лишь сведение схемы к задаче выполнимости 3-КНФ. Сведение за полиномиальное время называют *эффективным*.

Теорема. Задача выполнимости для схемы эффективно сводится к задаче выполнимости 3-КНФ.

Доказательство:

Введём переменные x_1, x_2, \dots, x_n , а еще добавим к ним g_1, g_2, \dots, g_s (которые у нас соответствуют элементам схемы). Тогда:

$$"g_i = \neg h_j" \Leftrightarrow (g_i \vee h_j) \wedge (\neg g_i \vee \neg h_j)$$

$$"g_i = h_k \vee h_l" \Leftrightarrow (h_k \vee h_l \vee \neg g_i) \wedge (\neg h_k \vee g_i) \wedge (\neg h_l \vee g_i)$$

$$"g_i = h_k \wedge h_l" \Leftrightarrow (\neg h_k \vee \neg h_l \vee g_i) \wedge (h_k \vee \neg g_i) \wedge (h_l \vee \neg g_i)$$

Для каждого g_i выписываем эквивалентную формулу и тогда 3-КНФ получается конъюнкцией всех этих выражений для $g_1, \dots, g_s + \wedge(g_s)$

Тогда $\exists \vec{x} : C(\vec{x}) \Leftrightarrow \exists \vec{x}, \vec{g} : K(\vec{x}, \vec{g}) = 1$.

Теперь осталось показать, почему собственно эти задачи будут действительно эквивалентны:

\Rightarrow В эту сторону почти очевидно. Если схема выполнима, существует набор переменных, при котором $g_s = 1$. Тогда просто вычислим g_1, \dots, g_s , подставив переменные из данного набора и из построения 3-КНФ видно, что итоговая конъюнкция будет равна 1.

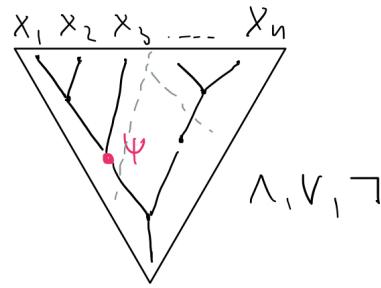
\Leftarrow В эту сторону тоже почти очевидно, но формально тут индукция по номеру элемента схемы. Отметим, что набор \vec{x} фиксирован. Нужно показать, что набор \vec{g} в КНФ соответствует значениям g_i в схеме. КНФ выполнима, т.е. каждый её конъюнкт равен 1. Посмотрим, например, на переменную g_1 : из того, как мы кодировали выражение для соответствующего ей элемента схемы, следует, что конъюнкт равен 1, если g_1 принимает верное значение равное значению операции над теми иксами, на которые ссылается. Это в точности и значит, что g_1 равно значению в схеме. Индуктивно показываем для всех остальные переменных g_i . Тогда при условии, что $g_s = 1$ получаем, что схема выполнима (прочувствуйте это).

Каждый элемент даёт константное число дизъюнктов (а в каждом не более 5 операций), поэтому сведение осуществляется за линейное время $\leq 15s$, где s - размер схемы, это $O(n)$.

12.2 Теорема о балансировке булевых функций

Теорема (о балансировке). Для формулы размера s существует формула глубины, не превышающей глубины $\leq 1 + 4 \log_2 s$

Доказательство:

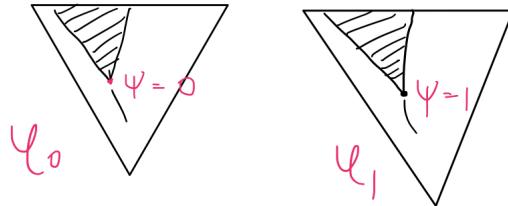


Формула представима в виде дерева (см. рисунок). Можно рассматривать подформулы (которые соответственно являются поддеревьями).

Лемма. $\exists \psi$: подформула с корнем ψ имеет размер $\geq \frac{s}{2}$, а в свою очередь любая её подформула имеет размер $< \frac{s}{2}$.

Доказательство леммы:

Доказательство этого утверждения совсем простое. Достаточно начать из выходного элемента и спускаться по ребрам формулы в подформулы так, чтобы каждый раз оставаться в подформуле размера $\leq \frac{s}{2}$. В выходном элементе размер подформулы равен s , а в переменных размер подформулы равен 0, так что в какой-то момент мы не сможем спуститься из очередной вершины в подформулу. Это как раз и будет означать, что у текущей подформулы размер не меньше $\frac{s}{2}$, а во всех ее подформулах уже меньше ■

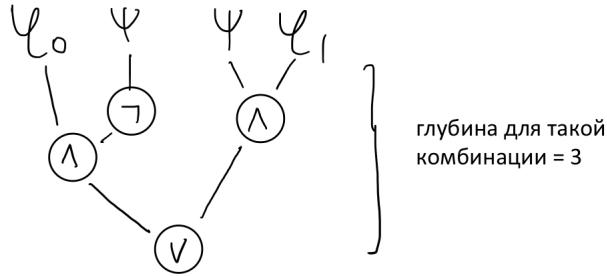


Вернёмся к док-ву самой теоремы. Воспользуемся индукцией по s . База очевидна верна для малых значений s . Будем действовать полной индукцией: для любой формулы размера меньше чем s утверждение верно. Находим ψ с соответствующим свойством по лемме. Важно, что каждый элемент в формуле используется один раз. Тогда сотрём все элементы в формуле кроме самой ψ . Создадим две новые формулы. В одной формуле зададим ψ константой 0, это будет φ_0 . Аналогично в φ_1 зададим ψ константой 1.

Описанная конструкция не соответствует нашим определениям: мы не разрешали использовать константы вместо переменных. Однако отрицание константы - константа, конъюнкция с 0 - то 0, с 1 - это второй аргумент конъюнкции. Аналогично дизъюнкция с 1 - это 1, с 0 - это второй аргумент дизъюнкции.

Пользуясь этими соображениями, в схеме с константами можно удалить лишние элементы. Ни размер схемы, ни глубина при таком удалении не увеличиваются. Тогда таким образом избавляясь от констант, какие-то константы могут спускаться ниже и в итоге получится формула без констант (или собственно константа, но это мы потом решим).

Рассмотрим тогда новую формулу $\alpha = (\varphi_0 \wedge \neg\psi) \vee (\varphi_1 \wedge \psi)$. Размер $\psi \geq \frac{s}{2}$, размер φ_0 и $\varphi_1 < \frac{s}{2}$, поэтому, применяя к ним предположение индукции, получаем, что для каждого из φ_0 и φ_1 существует формула глубиной $\leq 4 \log_2 \frac{s}{2} + 1 = 4(\log_2 s - 1) + 1 = 4 \log_2 s - 3$. Для того чтобы получить ψ нужно получить две подформулы ψ (потратив на это 1 глубины), причём для каждой подформулы ψ тоже верно предположение индукции. На этом шаге глубина уже $\leq 4 \log_2 s - 3 + 1 = 4 \log_2 s - 2$. И у общей формулы α глубина $\leq 4 \log_2 s - 2 + 3 = 4 \log_2 s + 1$ ■



13 Лекция 24

(Никита)

14 Семинар от 23.09.2016

Как обычно, начнём с разбора домашнего задания.

Задача 14.1. В ящике N различных шаров, из которых ровно M белых. Последовательно вынимают $n \leq N$ шаров. Пусть событие A_k означает, что k -й по счёту вынутый шар — белый, а событие B_m — что всего вынули $m \leq M$ белых шаров.

Найдите $\Pr(A_k | B_m)$, если (а) шары вынимаются без возвращения, (б) с возвращением.

Решение. Начнём со случая, когда нельзя возвращать шары. По определению условной вероятности $\Pr(A_k | B_m) = \frac{\Pr(A_k \cap B_m)}{\Pr(B_m)}$. Для начала посчитаем $\Pr(B_m)$. Как это сделать? Зафиксируем набор из n шаров, в котором первые m шаров белые. Какова вероятность того, что выпадет такой набор? Она равна

$$\frac{M}{N} \cdot \frac{M-1}{N-1} \cdot \dots \cdot \frac{M-m+1}{N-m+1} \cdot \frac{N-M}{N-m} \cdot \frac{N-M-1}{N-m-1} \cdot \dots \cdot \frac{N-M-(n-m)+1}{N-n+1}.$$

Теперь заметим, что если переставить числители местами, то получится вероятность того, что выпадет какой-то другой набор из n шаров, среди которых m белых. Тогда вероятность того, что выпадет хоть какой-то набор, подходящий под это условие, равна

$$\Pr(B_m) = \binom{n}{m} \frac{\frac{M!}{(M-m)!} \frac{(N-M)!}{(N-M-(n-m))!}}{\frac{N!}{(N-n)!}} = \frac{\binom{n}{m} \binom{N-n}{M-m}}{\binom{N}{M}}.$$

Теперь перейдём к числителю. Как посчитать $\Pr(A_k \cap B_m)$? В принципе, точно так же, как и $\Pr(B_m)$. Однако, в данном случае зафиксирована k -я позиция, поэтому нужно лишь выбрать $m-1$ позицию из $n-1$ для белых шаров. Тогда

$$\Pr(A_k \cap B_m) = \frac{\binom{n-1}{m-1} \binom{N-n}{M-m}}{\binom{N}{M}}.$$

Отсюда получаем, что $\Pr(A_k | B_m) = \frac{\binom{n-1}{m-1}}{\binom{n}{m}} = \frac{m}{n}$.

Переходим к случаю (б). Опять же, посчитаем $\Pr[B_m]$ и $\Pr[A_k \cap B_m]$. Рассуждения о перестановке так же имеют место, поэтому:

$$\Pr(B_m) = \binom{n}{m} \frac{M^m (N-M)^{n-m}}{N^n}$$

$$\Pr(A_k \cap B_m) = \binom{n-1}{m-1} \frac{M^m (N-M)^{n-m}}{N^n}$$

Подставляя полученные значения в формулу условной вероятности, получаем, что

$$\Pr(A_k | B_m) = \frac{\binom{n-1}{m-1}}{\binom{n}{m}} = \frac{m}{n}.$$

Ответ: $\frac{m}{n}$ в обоих случаях.

Задача 14.2. Ящик содержит a белых и b чёрных шаров (все шары различимы). Наудачу извлекается шар. Он возвращается обратно, и, кроме того, добавляется c шаров одного с ним цвета. Далее, подобная процедура повторяется снова. Пусть событие A_k означает, что на k -м шаге извлечён белый шар. Найдите

- (а) вероятность того, что при первых $n = n_1 + n_2$ извлечениях попалось n_1 белых и n_2 чёрных шаров;
- (б) вероятность события A_k ;
- (в) условную вероятность $\Pr(A_m | A_k)$ при $m > k$;
- (г) условную вероятность $\Pr(A_m | A_k)$ при $m > k$;

Решение. Рассмотрим ситуацию, когда последовательно выпало n_1 белых и n_2 чёрных шаров. Какова вероятность такого события? Она равна

$$\frac{a}{a+b} \cdot \frac{a+c}{a+b+c} \cdot \dots \cdot \frac{a+(n_1-1)c}{a+b+(n_1-1)c} \cdot \frac{b}{a+b+n_1c} \cdot \frac{b+c}{a+b+(n_1+1)c} \cdot \dots \cdot \frac{b+(n_2-1)c}{a+b+(n-1)c}.$$

Теперь переставим числители так, чтобы числители вида $a+x$ и $b+x$ были отсортированы по возрастанию. Тогда эта вероятность будет соответствовать какому-то другому набору из n_1 белых и n_2 чёрных. Пусть $\Pr(a, b, n_1, n_2)$ — вероятность того, что из a белых и b чёрных при первых $n = n_1 + n_2$ извлечениях попалось n_1 белых и n_2 чёрных шаров. Тогда

$$\Pr(a, b, n_1, n_2) = \binom{n}{n_1} \frac{a(a+c)\dots(a+(n_1-1)c)b(b+c)\dots(b+(n_2-1)c)}{(a+b)(a+b+c)\dots(a+b+(n-1)c)}.$$

Теперь посчитаем $\Pr(A_k)$. Пусть $C_{ki} = \{\text{до } k\text{-ой процедуры вытащили ровно } i \text{ белых шаров}\}$. Очевидно, что эти события образуют разбиение вероятностного пространства. Тогда по формуле полной вероятности $\Pr(A_k) = \sum_{i=0}^{k-1} \Pr(A_k \cap C_{ki})$.

Заметим, что $\Pr(A_k \cap C_{ki})$ совпадает с $\Pr(a, b, i+1, k-i-1)$ с тем лишь отличием, что в данном случае нужно выбрать i позиций из $k-1$:

$$\Pr[A_k \cap C_{ki}] = \binom{k-1}{i} \frac{a(a+c)\dots(a+ic)b(b+c)\dots(b+(k-i)c)}{(a+b)(a+b+c)\dots(a+b+(n-1)c)}.$$

Тогда

$$\Pr(A_k) = \frac{a}{a+b} \sum_{i=0}^{k-1} \binom{k-1}{i} \frac{(a+c)\dots(a+ic)b(b+c)\dots(b+(k-i)c)}{(a+b+c)\dots(a+b+(n-1)c)}.$$

Теперь заметим, что элемент суммы есть ни что иное, как $\Pr(a+c, b, i, k-i-1)$. Но

$$\sum_{i=0}^{k-1} \Pr(a+c, b, i, k-i-1) = 1,$$

так как эта сумма соответствует вероятности вытащить любой набор. Отсюда следует, что

$$\Pr(A_k) = \frac{a}{a+b}.$$

Перейдём к третьему (да и четвёртому тоже) пункту. По определению условной вероятности: $\Pr(A_k | A_m) = \frac{\Pr(A_k \cap A_m)}{\Pr(A_m)}$.

Как посчитать числитель? Точно так же, как и во втором случае. Пропустив аналогичные выкладки, выпишем ответ:

$$\Pr(A_k \cap A_m) = \frac{a(a+c)}{(a+b)(a+b+c)}.$$

Тогда получаем, что $\Pr(A_k | A_m) = \frac{a+c}{a+b+c}$.

Задача 14.3. Пусть A, B, C — попарно независимые равновероятные события, причём $A \cap B \cap C = \emptyset$. Найти максимально возможное значение $\Pr(A)$.

Решение. Начнём с того, что заметим следующее: $\Pr(A) \geq \Pr(A \cap (B \cup C)) = \Pr((A \cap B) \cup (A \cap C))$. Так как $A \cap B \cap C = \emptyset$, то $(A \cap B) \cap (A \cap C) = \emptyset$. Следовательно, $\Pr[A] \geq \Pr[A \cap B] + \Pr[A \cap C] = 2(\Pr[A])^2$ и $\Pr(A) \leq 1/2$.

Приведём пример, когда выполняется условие, причём $\Pr(A) = 1/2$. Рассмотрим классическую модель $\Omega = \{1, 2, 3, 4\}$ и события: $A = \{1, 2\}$, $B = \{2, 3\}$, $C = \{3, 4\}$. Легко понять, что данные события удовлетворяют условию и $\Pr(A) = 1/2$.

■

Задача 14.4. Игроки A и B играют в теннис. При розыгрыше на подаче A игрок A выигрывает с вероятностью p_1 , а при розыгрыше на подаче B — с вероятностью p_2 , все розыгрыши независимы. Игрок A подаёт первым, а выигрывает тот, кто первым наберёт n очков. Существует два варианта правил перехода подачи:

- (а) поочерёдная;
- (б) игрок подаёт до тех пор, пока не проиграет розыгрыш.

Покажите, что вероятность выигрыша A не зависит от правил перехода подачи, и вычислите её.

Решение. Будем считать, что всего было проведено $2n - 1$ розыгрышей. При таком количестве один игрок гарантированно наберёт не менее n очков, а второй — гарантированно меньше. Теперь опишем вероятностное пространство. Элементарные исходы будут иметь вид $\omega = (a_1, a_2, \dots, a_{2n-1})$, где $a_i \in \{0, 1\}$ (0 соответствует проигрышу, 1 — победе). Согласно этой схеме исход будет подходящим, если в наборе будет не меньше n единиц.

Начнём с пункта (а).

Поймём, как посчитать вероятность какого-либо элементарного случая. Пусть $n = 4$ и мы хотим найти вероятность элементарного исхода 0110110. Она равна $(1 - p_1)p_2p_1(1 - p_2)p_1p_2(1 - p_1) = p_1^2p_2^2(1 - p_1)^2(1 - p_2)^1$. Отсюда получаем закономерность: вероятность элементарного исхода $a_1a_2\dots a_{2n-1}$ равна

$$p_1^{\sum_{i=0}^{n-1} a_{2i+1}} p_2^{\sum_{i=1}^{n-1} a_{2i}} (1 - p_1)^{n - \sum_{i=0}^{n-1} a_{2i+1}} (1 - p_2)^{n - 1 - \sum_{i=1}^{n-1} a_{2i}}.$$

Пусть $k_1 = \sum_{i=0}^{n-1} a_{2i+1}$ — количество единиц на нечётных местах, а $k_2 = \sum_{i=1}^{n-1} a_{2i}$ — на чётных. Тогда вероятность того, что A выиграл, будет равна

$$\Pr = \sum_{\substack{k_1, k_2 \\ k_1+k_2 \geq n \\ k_1, k_2 \leq n}} \binom{n}{k_1} \binom{n-1}{k_2} p_1^{k_1} p_2^{k_2} (1 - p_1)^{n-k_1} (1 - p_2)^{n-1-k_2}.$$

Теперь перейдём к пункту (б).

Докажем следующее: между данными методами подачи есть биекция, т.е. игре с поочерёдной подачей можно сопоставить игру с подачей до проигрыша. Рассмотрим частный случай: поочерёдно вышел исход 0110110. Тогда можно “раскидать” партии так:

$$\begin{aligned} \text{Подаёт первый: } & 0110 \\ \text{Подаёт второй: } & 101 \end{aligned}$$

Биекция будет иметь вид 0101101. Как её построить? Разбиваем исход на подачи первого и второго игрока, тем самым получая строки длиной n и $n - 1$. После этого строим по ним новую строку по следующему алгоритму: [H] Построение исхода в случае подачи до проигрыша по исходу в случае поочерёдной подачи [1] Начинаем со строки длины n (строки для первого игрока); Копируем строку посимвольно до тех пор, пока не попадём на 0; Переходим на другую строку; Повторяем два предыдущих шага до тех пор, пока не перенесём все символы. Алгоритм построения исхода при поочерёдной подаче по исходу при подаче до проигрыша будет почти аналогичен. ■

Перейдём к задачам на тему математического ожидания и дисперсии.

Задача 14.5. Бросили два N -гранных кубика. Пусть ξ — сумма выпавших очков. Найдите $\mathbb{E}[\xi]$ и $\mathbb{D}[\xi]$.

Решение. Пусть $\mathbb{E}[\xi_1]$ — матожидание количества очков, выпавших на первом кубике. Посчитать его несложно: $\mathbb{E}[\xi_1] = \sum_{i=1}^N \frac{i}{N} = \frac{N+1}{2}$. Заметим, что $\mathbb{E}[\xi] = \mathbb{E}[\xi_1 + \xi_2] = \mathbb{E}[\xi_1] + \mathbb{E}[\xi_2]$. Тогда

$$\mathbb{E}[\xi] = \frac{N+1}{2} + \frac{N+1}{2} = N+1.$$

Дисперсию будем считать по следующей формуле: $[\xi_1] = \mathbb{E}[\xi_1^2] - (\mathbb{E}[\xi_1])^2$. Посчитаем первый член:

$$\mathbb{E}[\xi_1^2] = \sum_{i=1}^N \frac{i^2}{N} = \frac{N(2N+1)(N+1)}{6N} = \frac{(2N+1)(N+1)}{6}.$$

Отсюда получаем, что

$$[\xi_1] = \frac{N+1}{2} \left(\frac{2N+1}{3} - \frac{N+1}{2} \right) = \frac{(N+1)(N-1)}{12} = \frac{N^2-1}{12}.$$

Заметим, что ξ_1 и ξ_2 независимы (ведь кубики тоже независимы). Тогда $[\xi] = [\xi_1] + [\xi_2]$ и

$$[\xi] = \frac{N^2-1}{6}.$$

Задача 14.6. Пусть выбрана случайная перестановка $\sigma \in S_n$. Введём случайную величину ξ , равную количеству стационарных точек (чисел i таких, что $\sigma(i) = i$). Найдите $\mathbb{E}[\xi]$ и $[\xi]$.

Решение. Введём событие $A_i = \{\sigma(i) = i\}$. Тогда $\xi = \sum_{i=1}^n I_{A_i}$. По свойству линейности:

$$\mathbb{E}[\xi] = \mathbb{E}\left[\sum_{i=1}^n I_{A_i}\right] = \sum_{i=1}^n \mathbb{E}[I_{A_i}] = \sum_{i=1}^n \Pr(A_i)$$

Так как $\Pr(A_i) = \frac{(n-1)!}{n!} = \frac{1}{n}$, то $\mathbb{E}[\xi] = 1$.

Теперь перейдём к подсчёту дисперсии. Распишем дисперсию через ковариации:

$$[\xi] = (\xi, \xi) = \left(\sum_{i=1}^n I_{A_i}, \sum_{i=1}^n I_{A_i} \right) = \sum_{i=1}^n \sum_{j=1}^n (I_{A_i}, I_{A_j}).$$

Посчитаем (I_{A_i}, I_{A_j}) . По свойству ковариации она равна

$$\mathbb{E}[I_{A_i} I_{A_j}] - \mathbb{E}[I_{A_i}] \mathbb{E}[I_{A_j}] = \mathbb{E}[I_{A_i \cap A_j}] - \mathbb{E}[I_{A_i}] \mathbb{E}[I_{A_j}] = \Pr[A_i \cap A_j] - \Pr[A_i] \Pr[A_j].$$

Возникают два случая:

(a) $i = j$. Тогда $(I_{A_i}, I_{A_j}) = \frac{1}{n} - \frac{1}{n^2}$.

(б) $i \neq j$. Тогда $\Pr[A_i \cap A_j] = \frac{(n-2)!}{n!} = \frac{1}{n(n-1)}$ и $(I_{A_i}, I_{A_j}) = \frac{1}{n^2(n-1)}$.

Отсюда получаем, что

$$[\xi] = \frac{n(n-1)}{2} \frac{1}{n^2(n-1)} + n \left(\frac{1}{n} - \frac{1}{n^2} \right) = 1.$$