Дискретная математика

4 апреля 2019 г.

Содержание

1 Контакты преподавателя

Татьяна Викторовна Абрамовская, tanya.abramovskaya@gmail.com

2 Список литературы

- Романовский И.В. "Дискретный анализ"
- Иванов, Якубович "Введение в комбинаторику (теория и задачи)"

3 Теория множеств

3.1 01.09.18

Множество - это коллекция объектов произвольной природы. Обозначаются прописными латинскими либо греческими буквами.

Элементы множества - это объекты, составляющие множество.

Примеры: $\Phi = \{1, \lambda, element, \{IV, white\}\}$ - множество, состоящее из элементов $1, \lambda, element, \{IV, white\}$, причем последний элемент сам является множеством.
$1 \in \Phi$ - 1 содержится в Ф или 1 - элемент Ф
$IV \notin \Phi$ - IV не содержится в Ф, IV не является элементом Ф

Подмножество
A - произвольное множество
$B \subseteq A \Leftrightarrow \forall b \in B\ b \in A$,
то есть B называют подмножеством A тогда и только тогда, когда любой элемент множества B является также элементом множества A.

3.1.1 Собственное подмножество

$B \subseteq A$ и $\exists x \in A\ x \notin B \Leftrightarrow B \subset A$,
то есть собственным подмножеством множества A называют такое подмножество B, что в A существует элемент, который не является элементом B.
$B \subsetneqq A$ - такая запись может использоваться чтобы подчеркнуть, что B - собственное подмножество A.

3.1.2 Равенство множеств

Если $B \subseteq A$ и $A \subseteq B$, то $A = B$.

3.1.3 Мощность множества

- это число элементов в нем (для конечных множеств). Обозначается как $|A|$

3.1.4 Пустое множество

- множество, мощность которого равна 0. Обозначается как $\varnothing$

3.1.5 Способы задания множества

- Полное перечисление элементов, например $\{1, 2, 3, 4, 5\}$

- Интуиция. Запись $\{1, 2, \ ... \ , 10\}$ очевидно задает множество натуральных чисел от 1 до 10, хотя формально такая запись смысла не имеет.
  $\{1, 2, \ ...\}$ задает множество всех натуральных чисел (обозначается как $\mathbb{N}$. Преподаватель удваивает не поперечный штрих, а левый вертикальный, но делать так в Latex я не умею). Запись $n \in \mathbb{N}$ $\{1, 2, \ ... \ , n\}$ формально некорректна для $n = 1$, но интуитивно понятно, что в таком случае двойку надо выбросить. Множество натуральных чисел от m до n мы будем обозначать как $m : n$

- Условие выбора. $\{x \in \mathbb{N} : x \vdots 2\}$ задает множество всех натуральных чисел, которые делятся на 2, то есть $\{2, 4, \ ...\}$. Слева от двоеточия задается множество, откуда мы выбираем элементы, справа - условие выбора. Запись $\{x :< something >\}$ означает, что в множестве содержатся ЛЮБЫЕ объекты, которые удовлетворяют условию, записанному справа от двоеточия.

- Множества могут быть заданы как результат некоторых операций над множествами.

3.1.6 Операции над множествами

A, B - произвольные множества. || здесь и далее обозначает логическое ИЛИ, && - логическое И.

- Объединение множеств: $A \cup B = \{x : x \in A \ || \ x \in B\}$
- Пересечение множеств: $A \cap B = \{x : x \in A \ \&\& \ x \in B\}$
- Разность множеств: $A \setminus B = \{x : x \in A \ \&\& \ x \notin B\}$

Свойства операций над множествами
Объединение и пересечение обладают ассоциативностью и коммутативностью. Доказать это можно проверив соответствующие равенства (например, для ассоциативности объединения - $A \cup (B \cup C) = (A \cup B) \cup C$) по определению равенства множеств.

Запись
$n \in \mathbb{N}$ $A_1 \cup A_2 \cup ... \cup A_n = \bigcup\limits_{i \in 1:n} A_i$
$n \in \mathbb{N}$ $A_1 \cap A_2 \cap ... \cap A_n = \bigcap\limits_{i \in 1:n} A_i$

3.1.7 Множество всех подмножеств данного множества

Пусть A - произвольное множество. Тогда $\{B : B \subseteq A\} = 2^A$
$|2^A| = 2^{|A|}$ (доказательство этого факта интуитивно понятно, если знать, что такое битовые маски)

3.1.8 Примеры:

$\{element\} \subseteq 2^{\Phi}$ - ложно, так как $element \not\subseteq \Phi$
$\{element\} \in 2^{\Phi}$ - верно, так как $\{element\} \subseteq \Phi$
$\varnothing \in 2^{\Phi}$, так как $\varnothing$ является подмножеством любого множества.
$\Phi \in 2^{\Phi}$

3.1.9 Разбиение множества

A - произвольное множество. $\Lambda = \{\Lambda_1, \Lambda_2, \ ... \ , \Lambda_n\}$, где $n \in \mathbb{N}$ называется разбиением множества A, если:

- $\Lambda \subseteq 2^A$
- $\forall i \in 1 : n \ \Lambda_i \neq \varnothing$
- $\forall i, j \in 1 : n \ i \neq j \ \Lambda_i \cap \Lambda_j = \varnothing$
- $\bigcup\limits_{i \in 1:n} \Lambda_i = A$

3.1.10 Измельчение разбиения

A - произвольное множество, Λ и K - разбиения A, $|\Lambda| = m$, $|K| = n$. Λ называют измельчением K (или говорят, что Λ мельче K), если $\forall i \in 1 : m \ \exists j \in 1 : n \ \Lambda_i \subseteq K_j$
Стоит помнить, что Λ всегда мельче Λ

3.1.11 Произведение разбиений

A - произвольное множество, Λ и K - разбиения A, $|\Lambda| = m$, $|K| = n$. Произведением разбиений K и Λ называется такое разбиение Π множества A, которое мельче K и мельче Λ, и при этом самое крупное из этих измельчений (то есть, все разбиения, которые мельче K и мельче Λ будут также мельче Π).

Доказательство существования:
Определим Π_{ij} как $\Lambda_i \cap K_j$, Π_0 как множество всех Π_{ij} для $i \in 1 : m, j \in 1 : n$
и докажем, что Π является произведением Λ и K.
Докажем, что $\Pi = \{\Pi_0 : \Pi_{ij} \neq \varnothing\}$ является произведением Λ и K

- Докажем, что $\forall i \in 1 : m, \ \forall j \in 1 : n \ \Pi_{ij} \subseteq A$. Так как K и Λ - разбиения множества A, $\Lambda_i \subseteq A, K_j \subseteq A \Rightarrow \Pi_{ij} = \Lambda_i \cap K_j \subseteq A$.
- Докажем, что $\forall i, p \in 1 : m, \ \forall j, q \in 1 : n, \ i \neq p \ \Pi_{ij} \cap \Pi_{pq} = \varnothing$. $\forall x \in \Pi_{ij} \ x \in \Lambda_i$ по определению Π_{ij}. Так как Λ - разбиение множества A, $\Lambda_i \cap \Lambda_p = \varnothing \Rightarrow x \notin \Lambda_p \Rightarrow x \notin \Pi_{pq} \Rightarrow \Pi_{ij} \cap \Pi_{pq} = \varnothing$

- Аналогично докажем, что $\forall i, p \in 1:m,\ \forall j, q \in 1:n,\ j \neq q\ \Pi_{ij} \cap \Pi_{pq} = \varnothing$. Таким образом, мы доказали, что пересечение любых двух элементов Π - пустое множество.

- Докажем, что $\bigcup\limits_{i \in 1:m, j \in 1:n} \Pi_{ij} = A$. Во-первых, поскольку, как было доказано выше, $\forall i \in 1:m,\ \forall j \in 1:n\ \Pi_{ij} \subseteq A$, $\bigcup\limits_{i \in 1:m, j \in 1:n} \Pi_{ij} \subseteq A$. Во-вторых, поскольку K и Λ - разбиения множества A, $\forall a \in A \exists i \in 1; m, j \in 1:n\ a \in \Lambda_i, a \in K_j \Rightarrow a \in \Pi_{ij} \Rightarrow a \in \bigcup\limits_{i \in 1:m, j \in 1:n} \Pi_{ij} \Rightarrow A \subseteq \bigcup\limits_{i \in 1:m, j \in 1:n} \Pi_{ij}$, то есть по определению равенства множеств, $\bigcup\limits_{i \in 1:m, j \in 1:n} \Pi_{ij} = A$. Очевидно, что добавление или изъятие из списка объединяемых множеств любого количества пустых множеств никак не влияет на результат объединения.

- Π мельче Λ и мельче K по определению Π, так как любой элемент Π является подмножеством какого-то элемента Λ и подмножеством какого-то элемента K.

- Докажем, что если разбиение Ω множества A мельче Λ и мельче K, то оно мельче Π. Так как Ω мельче Λ и мельче K, то $\forall \omega \in \Omega\ \exists i \in 1:m\ \omega \subseteq \Lambda_i;\ \exists j \in 1:n\ \omega \subseteq K_j \Rightarrow \omega \subseteq \Pi_{ij}$. $\omega \neq \varnothing$ так как Ω - разбиение. Следовательно, $\Pi_{ij} \neq \varnothing \Rightarrow \Pi_{ij} \in \Pi$. Таким образом, мы доказали, что произведение произвольных разбиений Λ и K произвольного множества A существует.

3.2 14.09.18

3.2.1 Математическая индукция

Пусть $\Sigma \subseteq \mathbb{N}, 1 \in \Sigma$ (база индукции). Известно, что $1 : n \subseteq \Sigma \Rightarrow n+1 \in \Sigma$. Докажем, что из этого следует, что $\Sigma = \mathbb{N}$ и обратно.
Предположим обратное. Тогда найдем минимальное $k \in \mathbb{N}, k \notin \Sigma$. По определению Σ, $k \neq 1$. $1 : k-1 \subseteq \Sigma$, а значит, $k \in \Sigma$. Мы получили противоречие, значит, предположение было неверно, значит, $\Sigma = \mathbb{N}$.
В обратную сторону доказывается очевидно, ведь если $\Sigma = \mathbb{N}$, то $\forall n \in \mathbb{N}\ n+1 \in \Sigma$.
Замечание: в некоторых случаях удобнее переопределять индукцию так, чтобы базой была не единица, а ноль. Тогда вместо $\mathbb{N}$ будет $\mathbb{N}_0$.

3.2.2 Мощность 2^A

Докажем, что $\forall A\ |A| \in \mathbb{N}_0,\ |2^A| = 2^{|A|}$. Определим $\Sigma = \{n \in \mathbb{N}_0 : \forall A\ |A| = n,\ |2^A| = 2^{|A|}\}$
$0 \in \Sigma$, так как $|\varnothing| = 0, 2^{\varnothing} = \{\varnothing\}, |2^{\varnothing}| = 1 = 2^0 = 2^{|\varnothing|}$
Предположим, что $n \in \mathbb{N}_0\ \forall A\ |A| \in 0 : n,\ |2^A| = 2^{|A|}$. Тогда $\forall X\ |X| = n+1,\ |X| \geq 1 \Rightarrow \forall x \in X\ |X \setminus \{x\}| = n$.
Тогда можно представить 2^X как дизъюнктное объединение множества всех подмножеств X, не содержащих x и множества всех подмножеств X содержащих x. Тогда $|2^X| = |2^{X\setminus\{x\}}| + |\{Y \cup \{x\} : Y \in 2^{X\setminus\{x\}}\}|$.
$|2^{X\setminus\{x\}}| = 2^n$, так как $|X \setminus \{x\}| = n$
$|\{Y \cup \{x\} : Y \in 2^{X\setminus\{x\}}\}| = 2^n$
$|2^X| = 2^n + 2^n = 2^{n+1}$

3.2.3 Упорядоченная пара

A, B - произвольные множества, $a \in A, b \in B$. (a, b) - упорядоченная пара. $(a_1, b_1) = (a_2, b_2) \Leftrightarrow a_1 = a_2, b_1 = b_2$

3.2.4 n-арный упорядоченный кортеж

$A_1,\ ...\ A_n,\ B_1,\ ...\ B_m$.
$(a_1,\ ...\ a_n)$ - n-арный упорядоченный кортеж.
$(a_1,\ ...\ a_n) = (b_1,\ ...\ b_n) \Leftrightarrow m = n, \forall i \in 1 : n\ a_i = b_i$

3.2.5 Прямое (Декартово) произведение

Прямым произведением множеств A и B называют $A \times B = \{(a, b) : a \in A, b \in B\}$

3.2.6 Бинарное отношение

Бинарным отношением R на множествах A и B называют $R \subseteq A \times B$

3.2.7 Обратное бинарное отношение

Говорят, что R^{-1} - отношение обратное R, если $\forall (a, b) \in R\ (b, a) \in R^{-1}$, и наоборот, отношение, удовлетворяющее данному условию является обратным к R.

3.2.8 Отображение

Бинарное отношение R на множествах A и B называется отображением, если $\forall a \in A\ \exists! b \in B\ (a, b) \in R$

3.2.9 Сюръекция

Отображение R называется сюръективным, если $\forall b \in B\ \exists a \in A\ (a, b) \in R$

3.2.10 Инъекция

Отображение R называется инъективным, если $(a_1, b) \in R, (a_2, b) \in R \Rightarrow a_1 = a_2$

3.2.11 Биекция

Отображение R называется биективным, если оно одновременно сюръективно и инъективно

Пример:
$\{(\{a, \{a, b\}\}, (a, b)) : a \in A, b \in B\}$ - биекция.

3.2.12 n-арное отношение

$R \subseteq A_1 \times \ ... \ \times A_n$ - n-арное отношение на множествах $A_1, \ ... \ , A_n$.

3.2.13 Унарное отношение

$R \subseteq A$ - унарное отношение на множестве A

3.3 21.09.18

3.3.1 Свойства бинарных отношений

Бинарное отношение R над множеством A называется рефлексивным, если $\forall a \in A\ (a, a) \in R$
Бинарное отношение R над множеством A называется антирефлексивным, если $\forall a \in A\ (a, a) \notin R$
Бинарное отношение R над множеством A называется транзитивным, если $\forall a, b, c \in A\ (a, b) \in R,\ (b, c) \in R \Rightarrow (a, c) \in R$
Бинарное отношение R над множеством A называется антитранзитивным, если $\forall a, b, c \in A\ (a, b) \in R,\ (b, c) \in R \Rightarrow (a, c) \notin R$
Бинарное отношение R над множеством A называется симметричным, если $\forall a, b \in A\ (a, b) \in R \Rightarrow (b, a) \in R$
Бинарное отношение R над множеством A называется антисимметричным, если $\forall a, b \in A\ (a, b) \in R,\ (b, a) \in R \Rightarrow a = b$
Бинарное отношение R над множеством A называется асимметричным, если $\forall a, b \in A\ (a, b) \in R \Rightarrow (b, a) \notin R$

3.3.2 Отношение порядка

Бинарное отношение R над множеством A называется частичным порядком на A (или говорят, что A частично упорядочено R), если R рефлексивно, транзитивно и антисимметрично.
Бинарное отношение R над множеством A называется строгим частичным порядком на A, если R антирефлексивно, транзитивно и асимметрично.
Бинарное отношение R над множеством A называется линейным порядком на A, если R является частичным порядком на A и $\forall a, b \in A\ (a, b) \in R\ ||\ (b, a) \in R$
Бинарное отношение R над множеством A называется строгим линейным порядком на A, если R является строгим частичным порядком на A и $\forall a, b \in A\ (a, b) \in R\ ||\ (b, a) \in R$

3.3.3 Эквивалентность частичного порядка и отношения вложения

Произвольное множество A частично упорядочено R. Тогда существует биекция $f :\ A \to S$, где $S \subseteq 2^A$, $S = \{\{x \in A :\ (x, a) \in R\} :\ a \in A\}$.
Доказательство:
Определим $f(a) = \{x \in A :\ (x, a) \in R\}$
Сюръекция: по определению S.
Инъекция: предположим обратное. Тогда $\exists a, b \in A\ a \neq b,\ f(a) = f(b)$. $a \in f(a)$ и $b \in f(b)$ так как R рефлексивно. Значит, также $b \in f(a)$ и $a \in f(b)$. Но тогда по определению f $(a, b) \in R$ и $(b, a) \in R$, и при этом $a \neq b$, что противоречит антисимметричности R.
Таким образом, отношение частичного порядка R на множестве A эквивалентно отношению $\subseteq$ вложенности на множестве $S = \{\{x \in A :\ (x, a) \in R\} :\ a \in A\}$, так как $\forall a, b \in A\ (a, b) \in R \Rightarrow f(a) \subseteq f(b)$ и наоборот.

3.3.4 Минимальный/наименьший/максимальный/наибольший элемент

Произвольное множество A частично упорядочено R.
Минимальным элементом в A называют $m \in A$ такой, что $\nexists a \in A\ (a, m) \in R$. Минимальный элемент не обязательно единственный. В непустом конечном множестве существует хотя бы один минимальный элемент. (доказательство ниже).
Наименьшим элементом в A называют $m \in A$ такой, что $\forall a \in A\ (m, a) \in R$. Наименьший элемент не обязательно существует. Наименьший элемент единственный по предыдущему утверждению.
Максимальным элементом в A называют $m \in A$ такой, что $\nexists a \in A\ (m, a) \in R$. Минимальный элемент не обязательно единственный. В непустом конечном множестве существует хотя бы один максимальный элемент. (доказательство ниже).
Наибольшим элементом в A называют $m \in A$ такой, что $\forall a \in A\ (a, m) \in R$. Наибольший элемент не обязательно существует. Наибольший элемент единственный по предыдущему утверждению.
Доказательство существования минимального элемента в произвольном непустом конечном множестве A, частично упорядоченном отношением R:
Предположим обратное. Тогда $\forall a \in A\ \exists b \in A\ (b, a) \in R$. Тогда (поскольку A непусто) выберем любой элемент x из A и начнем строить цепочку $(x_1, x_2,\ ...)$ такую, что $x_1 = x$, а $\forall i \in N\ i > 1,\ x_i \in A\ (x_i, x_{i-1}) \in R$. По предположению эта цепочка имеет бесконечную длину и по транзитивности и антисимметричности R элементы в ней не повторяются, но поскольку A - конечное множество, а все элементы цепочки лежат в A, мы получаем противоречие.
Доказательство существования максимального элемента в произвольном непустом конечном множестве A, частично упорядоченном отношением R, аналогично.

3.3.5 Топологическая сортировка

Топологической сортировкой множества A, частично упорядоченного отношением R, называется такой линейный порядок T на A, что $(a, b) \in R \Rightarrow (a, b) \in T$.

3.3.6 1

$X \subseteq A \Rightarrow R(X)$ (сужение R на X) сохраняет все свойства R ((анти)рефлексивность, (анти)транзитивность, (а/анти)симметричность). Важно: эти свойства могут появиться при сужении, но не могут исчезнуть (то есть, например, сужение нерефлексивного отношения может быть рефлексивным, а вот сужение рефлексивного нерефлексивным - нет). Доказательство: в лоб проверить по определениям. Мне лень.

3.4 28.09.18

3.4.1 Лемма 2

В любом непустом конечном множестве A, частично упорядоченном отношением R существует минимальный элемент.
Доказательство: в пункте "Минимальный/наименьший/максимальный/наибольший элемент."предыдущей лекции.

3.4.2 Теорема о существовании топологической сортировки

A - произвольное конечное частично упорядоченное отношением R множество. Тогда на множестве A существует топологическая сортировка T, согласованная с R.
Доказательство:
Обозначим $A_0 = A$.
$\forall i \in \mathbb{N}_0, i \leq |A|$ если $A_i = \varnothing$, то его топологическая сортировка $T_i = \varnothing$
Если же A_i непустое, то по лемме 2 в нем существует минимальный элемент m_i. Тогда определим $A_{i+1} = A_i \setminus \{m_i\}, T_i = \{(m_i, a) : a \in A_i\} \cup T_{i+1}$. Докажем, что T_i является линейным порядком на A_i и согласовано с R (определение топологической сортировки), а значит, $T = T_0$ является топологической сортировкой $A_0 = A$.
Лемма *: если $(a, b) \in R$, то $\exists i, j \in 0 : (|A| - 1),\ i \leq j :\ a = m_i, b = m_j$. Доказательство: i и j существуют по построению T, $i \leq j$ так как если $j < i$, то $a \in A_j, a \neq b$, но тогда b не является минимальным элементом в A_j,так как $(a, b) \in R$. Стоит также заметить, что поскольку $i \leq j$, то $b \in A_i$, то есть, $(a, b) \in T$ (согласованность T с R).
Рефлексивность: по определению T_i.
Транзитивность: $\forall a, b, c \in A$, $(a, b) \in T$, $(b, c) \in T$, значит, по построению T $\exists i, j, k \in 0 : (|A| - 1)$ такие, что $a = m_i$, $b = m_j$, $c = m_k$, причем $i < j$ и $j < k$, значит, $i < k$, значит, $c \in A_i$, значит, $(a, c) \in T$.
Антисимметричность: Пусть $\exists a, b \in A\ :\ (a, b) \in T, (b, a) \in T$, значит, по построению T $\exists i, j \in 0 : (|A| - 1)$ такие, что $a = m_i$, $b = m_j$. Так как $(a, b) \in T$, $i \leq j$, а так как $(b, a) \in T$, $j \leq i$, то есть, $i = j$, то есть, $a = b$.
Линейный порядок:$\forall a, b \in A\ \exists i, j \in 0 : (|A| - 1) : a = m_i, b = m_j$. Если $i < j$, то $(a, b) \in T$, иначе $(b, a) \in T$ по построению T.
Согласованность: доказана в лемме *.

3.4.3 Цепь

Цепью на множестве A, (строго) частично упорядоченном отношением R, называют всякое $X \subseteq A$, линейно упорядоченное сужением R(X). Если цепь имеет наибольший элемент, говорят что цепь заканчивается в этом элементе.

3.4.4 Глубина элемента

Глубиной элемента a множества A, (строго) частично упорядоченного отношением R будем называть максимальную длину цепи, заканчивающейся в a. Глубина a может быть равна бесконечности.

3.4.5 Расписание

Расписанием на строго частично упорядоченном отношением R множестве A будем называть такое разбиение $(A_1, A_2, \ ... \ , A_n)$, что $\forall (a, b) \in R, \ b \in A_k \Rightarrow \ \exists j < k$, что $a \in A_j$

3.4.6 Теорема о кратчайшем расписании

A - конечное, строго частично упорядоченное отношением R множество, h - максимальная глубина элемента в A, $\forall i \in 1 : h \ A_i = \{a \in A : depth(a) = i\}$, тогда $\mathbb{A} = (A_1, A_2, \ ... \ , A_n)$ - кратчайшее расписание (расписание наименьшей мощности). Доказательство:
$\forall i \in 1 : h \ A_i \neq \varnothing$, так как: возьмем цепь $(z_1, \ ... \ , z_h)$ длины h (такая цепь существует, так как h - максимальная глубина элемента в A. $depth(z_h) = h$. $\forall i \in 2 : (h-1) \ depth(z_i) = i \Rightarrow depth(z_{i-1}) = i - 1$. Глубина не может быть меньше i - 1, так как глубина z_i равна i. Если же глубина z_{i-1} хотя бы i, то существует цепь, не содержащая (по транзитивности и антисимметричности R) z_i длины i, заканчивающаяся в z_{i-1}, но тогда к этой цепи можно дописать z_i и получить цепь длины $i + 1$, заканчивающуюся в z_i, что противоречит индукционному предположению о том, что глубина z_i равна i. Значит, $\forall i \in 1 : h \ A_i \neq \varnothing$
$\forall i \neq j \ A_i \cap A_j = \varnothing$, так как глубина элемента определяется единственным образом.
На данный момент, мы доказали, что $\mathbb{A}$ - разбиение. Теперь нужно доказать, что оно является расписанием.
Так как $\mathbb{A}$ - разбиение, $\forall a, b \in A \ \exists i : \ a \in A_i, \ \exists j : \ b \in A_j$
$(a, b) \in R \Rightarrow depth(a) < depth(b) \Rightarrow i < j$ по построению $\mathbb{A}$, то есть $\mathbb{A}$ - расписание.
Теперь докажем, что оно кратчайшее. Поскольку $|\mathbb{A}| = h$, то если $\mathbb{A}$ - не кратчайшее расписание, то должно существовать расписание мощности меньше h. Но поскольку h - глубина какого-то элемента в A, существует цепь длины h, а по определению расписания два элемента одной цепи не могут лежать в одном элементе расписания, следовательно, по принципу Дирихле, расписания с мощностью меньше h не существует, следовательно, $\mathbb{A}$ - кратчайшее расписание.

3.5 05.10.18

3.5.1 Антицепь

A - произвольное множество, строго частично упорядоченное отношением R. $X \subseteq A$ называют антицепью, если $\forall x, y \in X\ (x, y) \notin R(X)$

3.5.2 Разбиение на антицепи

Утверждение: всякое конечное частично упорядоченное множество высоты (высота множества - максимальная глубина элемента в нем) h можно разбить на h антицепей. Можно заметить, что кратчайшее расписание, существование которого конструктивно доказано в прошлой лекции, является как раз таким разбиением на антицепи.

3.5.3 Лемма Дилуорса

Во всяком конечном частично упорядоченном множестве A, $|A| = n\ \forall t \in 1 : n$ существует либо цепь длины $> t$, или антицепь длины $\geq \frac{n}{t}$.
Доказательство: предположим обратное. Тогда максимальная цепь в A имеет длину $\leq t$. Тогда A по предыдущему утверждению можно разбить на t антицепей. Пусть длина самой длинной из них равна l, $l < \frac{n}{t}$. Тогда в A не может быть больше чем $l * t$ элементов, но тогда получается, что $|A| = l * t < \frac{n}{t} * t = n$, что противоречит условию.

3.5.4 Последовательности

$S = 3, 2, 1, 8, 9, 4, 5, 6, 7$ - последовательность.
$S_1 = 2, 1, 4, 6$ - подпоследовательность.
$S_2 = 1, 4, 5, 6$ - возрастающая подпоследовательность.
$S_3 = 3, 2, 1$ - убывающая подпоследовательность.
$_S$" - бинарное отношение на S. $a <_S b$ тогда и только тогда, когда a предшествует b в последовательности S.
" $\prec$ " - бинарное отношение на S. $a \prec b \Leftrightarrow \{^{a <_S b}_{a < b}$
Цепь в S (частично упорядоченной $\prec$) - возрастающая подпоследовательность. Антицепь - убывающая подпоследовательность.

3.5.5 Возрастающая подпоследовательность наибольшей длины

Алгоритм: Начнем строить разбиение последовательности на антицепи (убывающие подпоследовательности) следующим образом: проходя по последовательности слева направо, будем добавлять каждый следующий элемент последовательности в минимальный по номеру элемент разбиения, в который можем. Ничерта не понятно? Вот пример:
Последовательность $3, 5, 8, 9, 4, 6, 1, 2, 7, 10$. Сначала добавим 3 в первый элемент разбиения, поскольку он пуст и нам ничего не мешает. Добавить в первый элемент 5 мы не можем, т.к. $3 \prec 5$, добавляем во второй. По тем же причинам, 8 идет жить в третий элемент разбиения, 9 - в четвертый. А вот 4 можно добавить во второй элемент разбиения, поскольку $3 \prec 4$, но $5 \not\prec 4$. Дальше 6 идет в третий элемент, 1 - в первый, 2 - во второй, 7 - в четвертый, а 10 - в пятый. Получаем разбиение: $\Lambda = \{\{3, 1\}, \{5, 4, 2\}, \{8, 6\}, \{9, 7\}, \{10\}\}$.
Теперь построим цепь: Обозначим $|\Lambda| = h$. $\forall i \in 2 : h\ \forall a \in \Lambda_i\ \exists prev(a)$ - элемент, лежащий в Λ_{i-1}

такой, что $prev(a) \prec a$ (если нет, то по алгоритму a бы попал в Λ_{i-1}). Таким образом, можно построить цепь A длины h таким образом: A_h - любой элемент Λ_h; $\forall i \in 1 : (h-1)\ A_i = prev(A_{i+1})$. Поскольку в S существует цепь длины h, S нельзя разбить менее чем на h антицепей (иначе два элемента из этой цепи окажутся в одной антицепи по принципу Дирихле), то есть разбиение Λ является минимальным. Поскольку разбиение Λ минимально, нельзя построить цепь длины большей чем h (иначе, опять-таки, два элемента этой более длинной цепи попадут в один элемент разбиения Λ по принципу Дирихле). Таким образом, мы одновременно научились строить самую длинную возрастающую подпоследовательность и минимальное разбиение на убывающие подпоследовательности.

3.5.6 Теорема Дилуорса

A - конечное частично упорядоченное отношением R множество. Доказать, что размер наибольшей антицепи Z равен размеру наименьшего разбиения на непересекающиеся цепи.
Доказательство: размер любой антицепи меньше либо равен количеству цепей в любом разбиении, так как иначе получится, что в такой антицепи, размер которой больше размера некоторого разбиения на цепи, два элемента этой антицепи будут лежать в одной цепи в разбиении.
Размер любой антицепи меньше либо равен размеру максимальной антицепи, количество элементов в наименьшем разбиении на цепи меньше либо равно количеству элементов в любом разбиении. Таким образом, получаем, что размер максимальной антицепи меньше либо равен количеству цепей в наименьшем разбиении. Притом получаем, что если размер какой-то антицепи равен размеру какого-то разбиения на цепи, то антицепь - максимальной длины, а разбиение - минимального размера. Осталось доказать строгое равенство.
База индукции: если $A = \varnothing$ или $R = \varnothing$, теорема очевидна. Если $|A| = 1$ - тоже.
Тогда рассмотрим случай $|A| \geq 2, R \neq \varnothing$. Тогда в A есть максимальный элемент M.
Индукционное предположение: для всех множеств, чья мощность меньше мощности A, теорема доказана. Индукционный переход: $A \setminus M$ разбивается на цепи $c_1,\ \ldots\ , c_k$, и притом в $A \setminus M$ существует хотя бы одна антицепь размера k.
Каждая антицепь размера k пересекается с каждой цепью разбиения по 1 элементу (иначе 2 элемента антицепи попадут в одну цепь).
Определим $\forall i \in 1 : k\ c_i \in C_i$ - максимальный элемент C_i, который лежит хотя бы в одной антицепи размера k. Докажем, что $Y = \{c1,\ \ldots\ , c_k\}$ - антицепь.
Зафиксируем для каждого c_i какую-нибудь антицепь Y_i размера k, которая содержит этот элемент (такие антицепи для разных c_i могут пересекаться). $\forall i \neq j \in 1 : k\ Y_i \cap C_j$ содержит ровно один элемент y_{ij}. $(y_{ij}, c_j) \in R$ в силу определения c_j.
$(c_j, c_i) \in R \Rightarrow (y, c_i) \in R$, чего быть не может, поскольку $c_i \in Y_i$ и $y_{ij} \in Y_i$, а Y_i - антицепь. Значит, $(c_j, c_i) \notin R$.
Аналогично рассмотрим $Y_j \cap C_i$ и получим, что $(c_i, c_j) \notin R$.
Таким образом, Y - антицепь размера k.
Теперь вспомним, что мы выкидывали из A M. Если $\exists i \in 1 : k\ (c_i, M) \in R$, то рассмотрим $X = \{x \in C_i\ :\ (x, c_i) \in R\} \cup M$. X, по построению, цепь.
$A \setminus X$ не содержит антицепи длины k, так как любая антицепь длины k должна иметь хотя бы один элемент в C_i, но в $C_i \setminus X$ по определению c_i нет элементов, которые входят в антицепь длины k. $A \setminus X$ содержит антицепь размера k - 1, а значит, по индукционному предположению, разбивается на k - 1 цепь. Тогда добавив к этому разбиению X, получим, что A разбивается на k цепей.

В случае же, если $\forall i \in 1 : k\ (c_i, M)\ /\!\!\int R\ ((M, c_i) \notin R$ так как M - максимальный элемент в A), $Y \cup M$ - антицепь размера k + 1. Больше быть не может, так как мы добавили только один элемент. В разбиение C_i должен добавиться один элемент. Не больше, так как существует разбиение $C_i \cup M$. Не меньше, так как размер разбиения на цепи не меньше чем размер любой антицепи, а у нас есть антицепь размера k + 1.
Таким образом, теорема Дилуорса доказана.

3.5.7 Альтернативное доказательство теоремы Дилуорса

Если $A = \varnothing$ или $R = \varnothing$, теорема очевидна. Если $|A| = 1$ - тоже. Если в R лежат только пары вида (x, x) - тоже.
Тогда рассмотрим случай $|A| \geq 2, R \neq \varnothing, \exists x \neq y \in A\ (x, y) \in R$.
В таком множестве можно выбрать минимальный элемент m и максимальный M такие, что $(m, M) \in R), m \neq M$. Почему?
(Упражнение) Потому что рассмотрим произвольные $x \neq y \in A, (x, y) \in R$. Если y - максимальный, выберем его как M, а иначе существует y_1 такой, что $(y, y_1) \in R$. Будем "увеличивать"y_i таким образом, пока не наткнемся на максимальный элемент y_{max} (а мы наткнемся, т.к. множество конечно). При этом очевидно, что $y_{max} \neq x$, иначе у нас сломается антисимметричность R между x и y. Обозначим $M = y_{max}$. Аналогично найдем и m.
Теперь рассмотрим $A \setminus \{m, M\}$. Пусть Y - наибольшей длины антицепь в A, $|Y| = s$.
Рассмотрим два случая:
1 случай - наибольшая антицепь в $A \setminus \{m, M\}$ имеет длину $s - 1$ ($s - 2$ не может быть, так как тогда в Y должны входить и m, и M, а $(m, M) \in R$). Тогда по индукционному предположению (да, мы все еще живем по индукции) получается, что $A \setminus \{m, M\}$ разбивается на s - 1 цепь. Добавляем к этому разбиению цепь $\{m, M\}$ и получаем разбиение размера s для A. Разбиения размера $s - 1$ для A быть не может, так как иначе максимальная антицепь в A будет иметь размер $s - 1$.
2 случай - наибольшая антицепь Z в $A \setminus \{m, M\}$ имеет длину s. Тогда определим:
$A^+ = \{a \in A\ :\ \exists z \in Z\ (a, z) \in R\}$
$A^- = \{a \in A\ :\ \exists z \in Z\ (z, a) \in R\}$
Докажем, что $A^+ \cap A^- = Z$. $Z \subseteq A^+ \cap A^-$ в силу рефлексивности R. $A^+ \cap A^- \subseteq Z$? Предположим обратное. Тогда $\exists z_\pm \in (A^+ \cap A^-) \setminus Z$ такое, что $\exists z_1, z_2 \in Z$ такие, что $(z_\pm, z_1) \in R$ и $(z_2, z_\pm) \in R$, но тогда по транзитивности R получается, что $(z_1, z_2) \in R$, а Z таки антицепь.
Докажем, что $A^+ \cup A^- = A$: Предположим обратное. Тогда $\exists z_\mp \in Z\ \forall z \in Z\ (z, z_\mp) \notin R, (z_\mp, z) \notin R$, но тогда $Z \cup \{z_\mp\}$ - цепь длины $s + 1$, что противоречит условию.
$m \notin A^-$, так как m - минимальный элемент в A, и при этом $m \notin Z$, так как Z мы строили в множестве $A \setminus \{m, M\}$, значит, $m \in A^+$
Аналогично, $M \in A^-$
A^+ и A^- содержат Z, а значит, каждая цепь в A^+ и A^- пересекает Z в одной точке, причем $\forall z \in Z$ цепь $C_z^+ \subseteq A^+, z \in C_z^+$ заканчивается в z, а цепь $C_z^- \subseteq A^-, z \in C_z^-$ в нем начинается, а значит, A разбивается на цепи вида $C_z^+ \cup C_z^-$, которых ровно s штук, поскольку каждая из этих цепей проходит через свой элемент Z.

4 Комбинаторика

4.1 12.10.18

4.1.1 Комбинаторное доказательство и пончики

Комбинаторное доказательство - способ нахождения мощности множества путем установления биекции с множеством известной мощности. Ничего не понятно? Пончики to the rescue!
Пусть у нас есть пончики 5 видов: с ванилью (В), шоколадом (Ш), карамелью (К), сгущенкой (С) и сахарной пудрой (П).
Пример 1: пусть мы хотим найти количество способов купить не более чем по одному пончику каждого вида. Тогда установим биекцию между множеством способов купить пончики и множеством двоичных кортежей длины 5. Для каждого кортежа вида $(0, 1, 0, 0, 1)$ скажем, что 0 выглядит как пончик, а потому определим, что если на i-й позиции в кортеже стоит 0, то мы купили пончик такого вида, а если 1 - то нет. То есть, в приведенном выше кортеже мы купим ванильный пончик, карамельный пончик и пончик со сгущенкой. Таким образом, мы построили биекцию (это, если очень хочется, можно по определению биекции проверить. А мне вот не хочется, мне хочется пончиков)между множеством способов купить не более чем по одному пончику каждого вида и множеством двоичных кортежей длины 5, мощность которого, как известно, равна $2^5 = 32$.
Пример 2: пусть мы хотим купить 12 пончиков, неважно какого вида. Покажем, что количество способов сделать это равно количеству двоичных кортежей длины 16, в которых ровно 4 единицы. Идея проста: нули в кортеже снова обозначают пончики, причем нули, идущие до i-й единицы отвечают за количество пончиков i-го вида (можно считать, что в конце кортежа стоит еще 17-й элемент, всегда равный единице, но поскольку он зафиксирован, на подсчет количества вариантов он не влияет). Таким образом, $(0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)$ соответствует варианту 2 В, 0 Ш, 5 К, 3 С, 2 П. Опять-таки, если вы еще не захлебнулись слюной, можете руками проверить, что это биекция. Ну а количество кортежей длины 16 с ровно 4 единицами равно C_{16}^4

4.2 19.10.18

4.2.1 Мощность множества всех подмножеств

A - произвольное конечное множество, $|A| = m$.
$\phi : A \to 1 : m$, если ϕ - биекция, то такое ϕ называется нумерацией элементов A.
Теперь во многих задачах мы можем вместо произвольных множеств мощности m использовать 1:m.
Рассмотрим $X = 1 : m$, $S \subseteq X$. Определим отображение $\chi : 2^X \to \{0,1\}^m$ из множества всех подмножеств X в множество всех n-арных двоичных кортежей следующим образом:
$\forall S \in 2^X \ \chi(S) = (\chi(S)_1, \ ... \ , \chi(S)_m)$, где $\chi(S)_i = \begin{smallmatrix}1, & i \in S \\ 0, & i \notin S\end{smallmatrix}$
χ по построению биекция (по подмножеству единственным образом строится кортеж, по кортежу единственным способом восстанавливается подмножество). Теперь построим отображение $\psi : \{0,1\}^m \to 0 : (2^m - 1)$ таким образом: $\psi(x) = \sum_{i=1}^{m} x_i * 2^{m-i}$, где $x = (x_1, x_2, \ ... \ , x_m)$. ψ тоже биекция (доказательство - перевод числа из десятичной системы в двоичную и обратно).
Таким образом, мы построили биекцию между A и 1:m, между $2^{1:m}$ и $\{0,1\}^m$, и наконец между $\{0,1\}^m$ и $0 : (2^m - 1)$. Тогда $|2^A| = |2^{1:m}| = \{0,1\}^m = |0 : (2^m - 1)| = 2^m$.

4.2.2 Перестановки

Перестановкой множества 1:n называется $< a_1, a_2, \ ... \ , a_n >$, где $\forall i \in 1 : n \ a_i \in 1 : n, \ \forall j \in (i+1) : n \ a_i \neq a_j$
Множество всех перестановок множества 1:n обозначается $< 1 : n >$.

4.2.3 Подсчет количества перестановок

Пусть $a \in < 1 : n >$ - перестановка. Тогда определим ее ключ $T(a) = (t_1^a, \ ... \ , t_n^a)$ следующим образом: $t_i^a = |\{j \in (i+1) : n \ : \ a_j < a_i\}|$, то есть, количество элементов перестановки, стоящих после i-го, которые при этом меньше i-го элемента перестановки. (Например, у перестановки $< 3, 1, 2 >$ ключ будет $(2, 0, 0)$).
Утверждение 1: $a, b \in < 1 : n >, a \neq b \Rightarrow T(a) \neq T(b)$.
Доказательство: так как перестановки различны, $\exists k \in 1 : n$ - наименьший индекс, в котором перестановки различаются $(a_k \neq b_k, \forall i \in 1 : (k-1) \ a_i = b_i)$. Не умаляя общности, скажем, что $a_k < b_k$ (иначе поменяем перестановки местами).
Так как $\{a_1, \ ... \ , a_{k-1}\} = \{b_1, \ ... \ , b_{k-1}\}$, $\{a_k, \ ... \ , a_n\} = \{b_k, \ ... \ , b_n\}$.
Следовательно, $\forall x \in \{a_{k+1}, \ ... \ , a_n\}, x < a_k \Rightarrow x \in \{b_{k+1}, \ ... \ , b_n\}, x < b_k$. Тогда, по построению T, $t_k^a \leq t_k^b$. Но при этом $a_k \in \{b_{k+1}, \ ... \ , b_n\}, a_k < b_k$, а значит, $t_k^a < t_k^b$, следовательно $T(a) \neq T(b)$.
Утверждение 2: $a, b \in < 1 : n >, T(a) \neq T(b) \Rightarrow a \neq b$.
Доказательство: Определим множество $\Pi^n = \{(\pi_1, \ ... \ , \pi_n) \ : \ \forall i \in 1 : n \ 0 \leq \pi_i \leq (n - i)\}$. Все ключи перестановок T(a) $\forall a \in < 1 : n >$ попадают в это множество по построению. Осталось показать, что по ключу можно единственным способом восстановить перестановку.
Прежде чем формализовать алгоритм восстановления, рассмотрим его на примере:
Пусть $T = (3, 6, 3, 2, 4, 3, 1, 0, 0)$. Первый элемент перестановки должен лежать в множестве $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, и в этом множестве должно быть ровно три элемента меньше него. Получается, на первом месте в перестановке должно стоять число 4. Теперь нужно найти в множестве оставшихся чисел $\{1, 2, 3, 5, 6, 7, 8, 9\}$ число такое, что в этом множестве есть ровно шесть чисел меньше его. Таким числом будет 8. Повторяя данный процесс, получим

перестановку $< 4, 8, 5, 3, 9, 7, 2, 1, 6 >$, причем поскольку все элементы перестановки различны, перестановку можно было восстановить единственным способом. Теперь формализуем алгоритм:
Дано: $T \in \Pi^n$
Задача: восстановить перестановку $a \in < 1 : n >$ такую, что $T(a) = T$. Определим множество p^i элементов множества 1:n, которые еще не получили свое место в перестановке ($p^1 = 1 : n$). Поскольку на каждом шаге алгоритма в перестановку будет добавляться один элемент, $|p^i| = n - i + 1$. Пронумеруем элементы p^i по возрастанию ($\forall k, j \in 1 : (n - i + 1), k < j \Rightarrow p^i_k < p^i_j$). Тогда получается, что элемент множества p^i такой, что в множестве p^i содержится ровно T_i элементов, меньших него, - это $p^i_{T_i+1}$. Тогда $a_i = p^i_{T_i+1}$. Множество элементов перестановки, стоящих в перестановке после a_i будет равно $p^{i+1} = p^i \setminus \{p^i_{T_i+1}\}$, и в нем будет ровно T_i элементов меньше a_i, то есть $T(a)_i = T_i$. Переходим к следующему шагу алгоритма.

4.2.4 Факториальная система счисления

Забивайте косяки, заваривайте алтайские травы, закидывайтесь грибочками, они вам пригодятся.
Итак, кроме привычных систем счисления с постоянным основанием (двоичная, десятичная, шестнадцатиричная и.т.д) существуют системы счисления со смешанным основанием. В данном случае, мы будем рассматривать факториальную систему счисления, то есть такую, что $n_{factorial} = \phi_m \phi_{m-1} \dots \phi_1$, $n_{10} = \sum_{i=1}^{m} \varphi_i * i!$ и $\forall i \in 1 : m \; 0 \le \varphi_i \le i$.
Утверждение: любое натуральное число представимо в факториальной системе счисления единственным образом.
Доказательство: Посмотрим на множество Π^n, которое мы определили в предыдущем пункте. С одной стороны, мы построили биекцию из него в множество $< 1 : n >$. С другой, очевидным образом строится биекция из Π^n в множество чисел, запись которых в факториальной системе счисления имеет меньше чем n знаков (выкинуть из любого $X \in \Pi^n$ ведущие нули и последний ноль - и мы получим некоторое число в факториальной системе счисления, в котором не более чем $n - 1$ знак. Обратно - выписать все цифры числа в факториальной системе счисления, дописать в конце ноль, а потом дописывать ведущие нули, пока не получим длину n).
Доказать, что все числа, запись которых в факториальной системе счисления имеет менее n знаков, меньше $n!$, можно по индукции:
Для n = 2 максимальным числом, в записи которого меньше n знаков, будет 1, что меньше $2! = 2$.
Если максимальное число m_n, в записи которого менее n знаков, меньше $n!$, то при переходе к n + 1 получим: $m_n + n! * n < n! * (n + 1) = n! * n + n!$.
Теперь докажем, что все числа, меньшие $n!$, представимы в факториальной системе счисления с менее чем n знаками.
Сначала у нас есть число $a < n!$. Обозначим $a_0 = a, n_0 = n$. Теперь найдем максимальное k_0 такое, что $k_0 * (n_0 - 1)! \le a_0$. $k_0 < n_0$, так как иначе $a_0 \le k_0 * (n_0 - 1)! \ge n_0!$. При этом, $a_1 = a_0 - k_0 * (n_0 - 1)! < (n_0 - 1)!$. Тогда можно перейти к $a_1 = a_0 - k_0 * (n_0 - 1)!, n_1 = n_0 - 1$. Алгоритм конечен, так как после $n - 1$ шага мы получим $a_{n-1} < n_{n-1}! = (n - (n - 1))! = 1! = 1$, а поскольку $\forall i \in 0 : (n - 1) \; a_i \ge 0$, $a_{n-1} = 0$, тогда $k_0 \dots k_{n-1}$ - корректное представление a в факториальной системе счисления.

Таким образом, мы построили биекцию между множеством $0 : (n! - 1)$ и множеством чисел, запись которых в факториальной системе счисления имеет не более n знаков.
Тогда для любого натурального m получаем, что поскольку существует некоторое натуральное n такое, что $n! > m$, m представимо в факториальной системе счисления.

4.2.5 Нумерация перестановок и перестановка, следующая за данной

В прошлом пункте мы построили биекцию из множества $< 1 : n >$ через множество чисел, имеющих не более n знаков в факториальной системе счисления, в множество $0 : (n! - 1)$, то есть пронумеровали перестановки из n элементов. Чтобы понять, как устроена эта нумерация, давайте попробуем по перестановке найти следующую за ней.
Пример:
$a =< 3, 8, 7, 6, 2, 4, 9, 5, 1$
$T(a) = (2, 6, 5, 4, 1, 1, 2, 1, 0)$
Представление в факториальной системе счисления выглядит так: $26541121_{factorial}$. Прибавляем единицу: $26541121_{factorial} + 1_{factorial} = 26541200$ (первый разряд переполнился, т.к. $1+1 = 2 > 1$, второй тоже, так как $2 + 1 = 3 > 2$, третий не переполнился).
Теперь по получившемуся числу восстановим перестановку b, следующую за a.
$T(b) = (2, 6, 5, 4, 1, 2, 0, 0, 0)$.
$b =< 3, 8, 7, 6, 2, 5, 1, 4, 9 >$.
Посмотрим, что изменилось в перестановке: "убывающий хвост"(максимальное количество элементов в конце перестановки, идущих в порядке убывания) перестановки и еще один элемент левее изменят свое положение, так как все разряды, соответствующие "убывающему хвосту переполнятся, а к следующему разряду прибавится единица. Таким образом, если длина убывающего хвоста была равна l, а $T(a)_{n-l} = k$, то последний $l + 1$ элемент T(b) будет выглядеть так: $(k + 1, 0, \ ... \ , 0)$. С точки зрения перестановки, это значит, что:

- первые $n - l - 1$ элементов перестановки останутся неизменны.
- на $(n - l)$-й позиции в b будет стоять минимальный элемент x множества $\{a_j : \ j \geq n - l\}$, такой что $x > k$.
- оставшиеся l элементов множества $\{a_j : \ j \geq n-l\} \setminus \{x\}$ будут стоять в конце перестановки b в порядке возрастания.

Пример: $a =< 5, 4, 1, 3, 2 >$, n = 5
Убывающий хвост a - $(3, 2)$ - имеет длину 2.
$(n - 2)$-й элемент a равен 1.
Минимальный элемент множества $\{a_j : \ j \geq n - 2\} = \{1, 2, 3\}$, больший 1, равен 2.
Таким образом, перестановка, следующая за a, имеет вид $< 5, 4, 2, 1, 3 >$

4.3 26.10.18

4.3.1 Подсчет подмножеств размера k

Пусть A - произвольное конечное множество, $|A| = n$. $S = \{M \subseteq A \ : \ |M| = k\}$. Чему равна мощность S?
$A = \{a_1, \ ... \ , a_n\}$
$\forall a_i \in < A > \ a =< a_{i_1}, \ ... \ , a_{i_k}, a_{i_{k+1}}, \ ... \ , a_{i_n} >$
$M = \{a_{i_1}, \ ... \ , a_{i_k}\}$. Так как $M \subseteq A$ и $|M| = k$, $M \in S$.
$A \setminus M = \{a_{i_{k+1}}, \ ... \ , a_{i_n}\} = \overline{M}$
$< a_{i_1}, \ ... \ , a_{i_k} > \in < M >$
$< a_{i_{k+1}}, \ ... \ , a_{i_n} > \in \overline{M}$
Таким образом, $\forall M' \in S$ можно установить биекцию $f : < M' > \times < \overline{M'} > \to < A >$, определенную следующим образом: $\forall b \in < M' > \ forallc \in < \overline{M'} > \ f((b, c)) =< b_1, \ ... \ , b_k, c_1, \ ... \ , c_{n-k} >$. А если без заумных формул, то это значит, что любую перестановку можно разрезать на перестановку некого подмножества размера k (первые k элементов перестановки) и некую перестановку его дополнения (остальные элементы).
Таким образом, комбинаторно получаем равенство $|S| * k! * (n - k)! = n!$ (количество перестановок множества A равно количеству способов выбрать подмножество A размера k, затем упорядочить его элементы и затем упорядочить элементы дополнения этого подмножества в множестве A). Отсюда получаем формулу для искомого $|S|$:
$|S| = \frac{n!}{k! * (n-k)!}$. Для таких чисел существует специальное обозначение C_n^k (в англоязычной литературе - $\binom{n}{k}$)

4.3.2 Разбиение на подмножества фиксированного размера

Заметим, что в прошлом пункте, выбрав подмножество размера k, мы автоматически выбрали и его дополнение размера $n-k$, таким образом построив разбиение A на два упорядоченных (не внутри себя, а в том плане, что если k окажется равно n - k, выбранные подмножества нельзя будет поменять местами. Формулировка не очень удачная, но на лекции было так) подмножества фиксированного размера (k и $n - k$). Количество таких разбиений, соответственно, - C_n^k.
Теперь обобщим эту задачу:
Сколькими способами можно разбить множество A мощности n на m упорядоченных (все в том же смысле, внутри подмножеств никакого порядка нет) подмножеств A_i, $|A_i| = k_i$, $\sum\limits_{i=1}^{m} k_i = n$?
Действуя по аналогии с предыдущей задачей, разрезаем перестановку длины n на m перестановок соответствуюших длин, получаем биекцию (благодаря тому, что мы умеем нумеровать элементы множеств, я могу заменить в записи подмножество размера k_i на $1 : k_i$, что позволяет не вводить 100500 обозначений) $< 1 : k_i > \times \ ... \ \times < 1 : k_n > \to < A >$, получив равенство
$P * k_1! * \ \ ... \ \ * k_m! = n!$, где P - искомое число разбиений множества A на m упорядоченных (нутыпонел) подмножеств фиксированных размеров. Отсюда $P = \frac{n!}{\prod\limits_{i=1}^{m} k_i!}$. Такие числа P обозначают как $C_n^{k_1, \ ... \ , k_m}$ или $\binom{n}{k_1, \ ... \ , k_m}$
Пример: давайте посчитаем количество анаграмм (перестановок букв) слова ПАРАЛЛЕЛЕПИПЕД. Каждой анаграмме можно сопоставить набор множеств позиций на которых стоят конкретные буквы, то есть для слова ПАРАЛЛЕЛЕПИПЕД получим для А множество $\{2, 4\}$, для Е - $\{7, 9, 13\}$ и так далее. Нетрудно заметить, что объединением таких множеств для всех

букв, содержащихся в слове, будет $1 : n$, где n - длина слова, эти множества можно упорядочить (например, упорядочив по алфавиту соответствующие им буквы) и эти множества не пересекаются (на одной позиции в слове не могут стоять две буквы сразу). Таким образом, количество анаграмм слова ПАРАЛЛЕЛЕПИПЕД равно отношению факториала 14 (длина слова) к произведению факториалов количества вхождений буквы в слово для каждой буквы, содержащейся в слове (2 для А, 3 для Е, 1 для И и.т.д.). Получаем $\frac{14!}{2!3!1!1!3!3!1!} = \frac{14!}{2!3!3!3!}$

4.3.3 Свойства C_n^k

$C_n^k = C_n^{n-k}$
Нетрудно заметить, что выбрав подмножество размера k множества размера n, мы однозначно выбрали и его дополнение - множество размера $n - k$.

$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$
Зафиксируем какой-то элемент a исходного множества. Мы можем либо взять его в наше подмножество размера k, и тогда нам останется выбрать оставшиеся $k - 1$ элементов из $n - 1$ других элементов множества, либо не брать, и тогда нам нужно набрать все k элементов из других $n - 1$ элементов исходного множества, а поскольку множество способов выбрать подмножество размера k, содержащее a, и множество способов выбрать подмножество размера k, не содержащее a, не пересекаются, мощность их объединения равна сумме их мощностей, то есть $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$

$C_{3n}^n = \sum_{r=0}^{n} C_n^r * C_{2n}^{n-r}$
Пусть в некой карточной игре карты делятся на карты существ и карты заклинаний. Пусть у абстрактного Вадима есть в коллекции n заклинаний и 2n существ, и ему нужно составить колоду размера n. Он мог бы просто наугад напихать туда все подряд одним из C_{3n}^n способов, не разбирая, где существа, а где заклинания. Но абстрактный Вадим умен, и он сперва хочет определить тактику, зафиксировав количество r заклинаний в колоде. Для конкретного r Вадим может составить колоду $C_n^r * C_{2n}^{n-r}$ способами. Нетрудно заметить, что r может принимать только значения от 0 до n (потому что у Вадима всего n заклинаний), и сумма количества способов составить колоду с r заклинаниями для всех r равна общему количеству способов составить колоду из n карт из 3n карт коллекции, то есть $C_{3n}^n = \sum_{r=0}^{n} C_n^r * C_{2n}^{n-r}$

4.3.4 Боль, унижение и формула включений и исключений

Если в какой-то момент приводимые здесь рассуждения становятся непонятными, попробуйте порисовать диаграммы Эйлера-Венна. Если все еще непонятно, напишите мне. Если вы не знаете, что такое диаграмма Эйлера-Венна, добро пожаловать в Википедию. Попробуем найти мощность объединения двух множеств. Если эти множества дизъюнктны (то есть, их пересечение пусто), то мощность объединения равна сумме мощностей. Если же пересечение непусто, то лежащие в нем элементы мы посчитали дважды, соответственно, из суммы мощностей надо вычесть мощность пересечения. $|A \cup B| = |A| + |B| - |A \cap B|$.

Для трех множеств будем рассуждать аналогично. $|A| + |B| + |C|$ учитывает дважды элементы, лежащие в попарных пересечениях. Но если вычесть мощности попарных пересечений ($|A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C|$), получится, что пересечение всех трех множеств мы учли три раза и вычли три раза (поскольку оно содержится как в каждом из множеств, так и в каждом из попарных пересечений), и нужно добавить его мощность к ответу: $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$.
Попробуем обобщить: мощность объединения n множеств $|A_1 \cup \ ... \ \cup A_n|$ - это сумма, в которую со знаком + входят все мощности пересечения нечетного количества множеств и со знаком − - все мощности пересечения четного количества множеств, то есть
$|A_1 \cup \ ... \ \cup A_m| = \sum\limits_{1 \le j \le n} (-1)^{j+1} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le n} |A_{i_1} \cap \ ... \ \cap A_{i_j}|$
Доказывать будем по индукции. База индукции: для n = 1 и n = 2 формулу мы уже рассмотрели.
Переход: Предположим, что для $n = m$ формула уже доказана, докажем ее для $n = m + 1$. воспользуемся ассоциативностью объединения
$|A_1 \cup \ ... \ \cup A_m \cup A_{m+1}| = |(A_1 \cup \ ... \ \cup A_m) \cup A_{m+1}|$
Теперь воспользуемся формулой для n = 2:
$|(A_1 \cup \ ... \ \cup A_m) \cup A_{m+1}| = |(A_1 \cup \ ... \ \cup A_m)| + |A_{m+1}| - |(A_1 \cup \ ... \ \cup A_m) \cap A_{m+1}|$
Теперь воспользуемся дистрибутивностью пересечения:
$|(A_1 \cup \ ... \ \cup A_m) \cup A_{m+1}| = |(A_1 \cup \ ... \ \cup A_m)| + |A_{m+1}| - |(A_1 \cap A_{m+1}) \cup \ ... \ \cup (A_m \cap A_{m+1})|$
Теперь можно воспользоваться формулой для n = m: $|(A_1 \cup \ ... \ \cup A_m) \cup A_{m+1}| = |(A_1 \cup \ ... \ \cup A_m)| + |A_{m+1}| - \sum\limits_{1 \le j \le n} (-1)^{j+1} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le m} |(A_{i_1} \cap A_{m+1}) \cap \ ... \ (\cap A_{i_j} \cap A_{m+1})|$
Заметим, что нам достаточно всего один раз пересечь с A_{m+1}, а также внесем минус под скобки:
$|(A_1 \cup \ ... \ \cup A_m) \cup A_{m+1}| = |(A_1 \cup \ ... \ \cup A_m)| + |A_{m+1}| + \sum\limits_{1 \le j \le n} (-1)^{j+2} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le m} |A_{i_1} \cap \ ... \ \cap A_{i_j} \cap A_{m+1}|$
Теперь посмотрим на эту формулу внимательно: все мощности пересечений, не содержащих A_{m+1} входят в первое слагаемое со знаком $(-1)^{j+1}$, где j - количество множеств в пересечении. $|A_{m+1}|$ - второе слагаемое - имеет знак $(-1)^0 = (-1)^2 = (-1)^{1+1}$, то есть -1 в степени "количество множеств в пересечении плюс один". Наконец, все пересечения, содержащие более одного множества и содержащие A_{m+1} входят в последнее слагаемое со знаком $(-1)^{j+2}$, где j - количество множеств в пересечении, не считая A_{m+1}, то есть со знаком $(-1)^{j'+1}$, где j' - количество множеств в пересечении. Таким образом, получается, что
$|A_1 \cup \ ... \ \cup A_m| = |(A_1 \cup \ ... \ \cup A_m)| + |A_{m+1}| + \sum\limits_{1 \le j \le n} (-1)^{j+2} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le m} |A_{i_1} \cap \ ... \ \cap A_{i_j} \cap A_{m+1}| =$
$\sum\limits_{1 \le j \le n} (-1)^{j+1} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le n} |A_{i_1} \cap \ ... \ \cap A_{i_j}|$
Формула включений и исключений доказана.

4.4 02.11.18

4.4.1 Подсчет количества разбиений, числа Белла и числа Стирлинга 2-го рода

Пусть есть произвольное конечное множество A, $|A| = n$, нужно подсчитать количество его разбиений мощности k (то есть, таких наборов множеств X_1, ... ,X_k, что $\forall i \in 1 : k\ X_i \neq \varnothing, \forall j \in 1 : k, i \neq j \Rightarrow X_i \cap X_j = \varnothing$).
Для $k = 2$ все довольно просто: нужно выбрать элементы, которые пойдут в первое множество разбиения, а все остальные, естественно, пойдут во второе. Таким образом, количество разбиений мощности 2 равно количеству способов выбрать непустое собственное (то есть, не совпадающее со всем множеством) подмножество A пополам (так как мы не упорядочиваем элементы разбиения, а значит, случаи $\{X_1, X_2\}$ и $\{X_2, X_1\}$ мы различать не должны), то есть таких способов $\frac{|2^A|-2}{2} = \frac{2^n-2}{2} = 2^{n-1} - 1$
Теперь перейдем к общему случаю:
Обозначим $B(n)$ число разбиений множества мощности n, $S(n, k)$ - число разбиений мощности k множества мощности n.
Нетрудно понять, что $B(n) = \sum_{i=1}^{k} S(n, k)$.
$B(n)$ называют n-ным числом Белла, а $S(n, k)$ - числом Стирлинга второго рода из n по k.
Примечание: поскольку числа Стирлинга второго рода встречаются намного чаще чем числа Стирлинга первого рода, я буду называть их просто числами Стирлинга. Алсо, в англоязычной литературе эти числа обозначаются $\{{n \atop k}\}$
Въетнамский флэшбек: как вы помните, у нас уже были числа C_n^k, и для них была рекуррентная формула $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}, C_n^0 = 1$. Теперь попробуем запилить похожую формулу для чисел Стирлинга.
Утверждается, что эта формула имеет вид $S(n, k) = S(n-1, k-1) + k * S(n-1, k)$
Доказательство:
Произвольное множество A мощности n $A = \{a_1, a_2, \ ... \ , a_n\}$
Для каждого разбиения этого множества верно одно из двух:

- либо $\{a_n\}$ - отдельный элемент разбиения, таких разбиений столько же, сколько разбиений мощности $k-1$ множества мощности $n-1$, то есть $S(n-1, k-1)$

- либо $\exists i \in 1 : k\ a_n \in X_i, |X_i| > 1$. Но тогда можно сначала найти разбиение мощности k множества $A \setminus \{a_n\}$ мощности $n-1$ (способов это сделать $S(n-1, k)$), а потом добавить a_n в один из элементов разбиения (k способов), то есть всего разбиений, в которых $\exists i \in 1 : k\ a_n \in X_i, |X_i| > 1$, $k * S(n-1, k)$

Поскольку очевидно, что множество разбиений, подпадающих под первый случай, и множество разбиений, подпадающих под второй случай, не пересекаются, то мощность их объединения равна сумме их мощностей, то есть
$S(n, k) = S(n-1, k-1) + k * S(n-1, k)$
Свойства чисел Стирлинга:
Во-первых, несколько крайних значений, на которые будет опираться наша формула:
$S(n, 0) = 0$

$S(0,0) = 0$
$S(k,n) = 0$ при $k > n$
Ну и пара значений чисто чтобы было:
$S(n,2) = 2^{n-1} - 1$
$S(n,n-1) = C_n^2$
Алсо, если кому не лень, можно порисовать треугольник Паскаля для чисел Стирлинга. Мне лень.

4.4.2 Рекуррентная формула для чисел Белла

$B(n+1) = \sum\limits_{k=0}^{n} C_n^k * B(k)$
Доказательство:
Пусть множество $A = \{a_1, \ ... \ , a_n, a_{n+1}\}$, $A = X_1 \cup \ ... \ \cup X_k$ - произвольное разбиение
$\exists i \in 1 : k \ a_{n+1} \in X_i$. $1 \leq |X_i| = j \leq (n+1)$
$|A \setminus X_i| = n + 1 - j$
Количество способов набрать X_i равно $C_n^{j-1} = C_n^{n-(j-1)} = C_n^{n+1-j}$, количество разбиений $A \setminus X_i$ - $B(n+1-j)$.
То есть, $B(n+1) = \sum\limits_{j=1}^{n+1} C_n^{n+1-j} * B(n+1-j)$.
Ну и чтобы получить исходную формулу, осталось только изменить нумерацию, приняв $k = n + 1 - j$:
$B(n+1) = \sum\limits_{k=0}^{n} C_n^k * B(k)$

4.4.3 Явная формула для чисел Стирлинга

Единственное, что меня беспокоило — это явные формулы. В мире нет никого более беспомощного, безответственного и безнравственного, чем человек, пытающийся вывести явную формулу. И я знал, что довольно скоро мы в это окунёмся. (C)
$\forall n \geq 0, k \geq 1 \ S(n,k) = \frac{1}{k!} \sum\limits_{j=0}^{k} (-1)^j * C_k^j * (k-j)^n$
Доказательство:
$S(0,k) = 0$. Для $n = 0$ формула имеет вид:
$S(n,k) = \frac{1}{k!} \sum\limits_{j=0}^{k} (-1)^j * C_k^j * (k-j)^0 =$
$= \frac{1}{k!} \sum\limits_{j=0}^{k} (-1)^j * C_k^j =$
$= \frac{1}{k!} \sum\limits_{j=0}^{k} (-1)^j * C_k^j * 1^{k-j} =$(в этот момент мы видим формулу бинома Ньютона)
$= \frac{1}{k!}(1 + (-1))^k = 0$
А теперь поехали обобщать. $L = \{R \subseteq A \times 1 : k \ : \ R$ - сюръекция$\}$ - множество сюръективных отображений из A в 1:k. Нетрудно понять, что это множество равномощно множеству упорядоченных (то есть, где X_i и X_j нельзя менять местами) разбиений мощности k множества A (если $R(a) = i$, то в разбиении $a \in X_i$ и наоборот).
Чтобы получить наши родные неупорядоченные разбиения достаточно поделить на $k!$, то есть

$S(n,k) = \frac{1}{k!} * |L|$.
Теперь найдем мощность L. Для этого найдем мощность множества всех отображений из A в 1:k, а потом вычтем мощность множества всех несюръективных отображений из A в 1:k.
Обозначим множество всех отображений из A в 1:k как M. $|M| = k^n$
Теперь обозначим для $\forall i \in 1:k$ $P_i = \{R \subseteq A \times 1:k \ : \ R$ - отображение, $\nexists a \in A \ R(a) = i\}$
Тогда $|L| = |M| - |R_1 \cup \ ... \ \cup R_k| = k^n - |R_1 \cup \ ... \ \cup R_k|$
$|R_1 \cup \ ... \ \cup R_k|$ найдем по формуле включений и исключений:
$|R_1 \cup \ ... \ \cup R_k| = \sum_{j=1}^{k} (-1)^{j+1} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le k} |P_{i_1} \cap \ ... \ \cap P_{i_j}|$
$|P_{i_1} \cap \ ... \ \cap P_{i_j}| = (k-j)^n$ - количество отображений из A в $1:k \setminus \{i_1, \ ... \ , i_j\}$ - не зависит от выбора конкретных $\{i_1, \ ... \ , i_j\}$, а значит, $\sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le k} |P_{i_1} \cap \ ... \ \cap P_{i_j}| = C_k^j * (k-j)^n$, поскольку выбрать множество $\{i_1, \ ... \ , i_j\}$ из $1:k$ можно C_k^j способами. Тогда получаем:
$|R_1 \cup \ ... \ \cup R_k| = \sum_{j=1}^{k} (-1)^{j+1} \sum\limits_{1 \le i_1 \le \ ... \ \le i_j \le k} |P_{i_1} \cap \ ... \ \cap P_{i_j}| = \sum\limits_{j=1}^{k} (-1)^{j+1} * C_k^j * (k-j)^n$
Откуда
$|L| = k^n - \sum\limits_{j=1}^{k} (-1)^{j+1} * C_k^j * (k-j)^n = k^n + \sum\limits_{j=1}^{k} (-1)^j * C_k^j * (k-j)^n = (-1)^0 * C_k^0 * (k-0)^n + \sum\limits_{j=1}^{k} (-1)^j *$
$C_k^j * (k-j)^n = \sum\limits_{j=0}^{k} (-1)^j * C_k^j * (k-j)^n$
Явная формула для чисел Стирлинга доказана.

5 Теория вероятности

ВИРАЯТНАСТЬ! ВИРАЯЯЯЯЯТНАААСТЬ!!! (С)

5.1 09.11.18

5.1.1 Вероятностное пространство

Пусть S - конечное множество, $|S| = n$.
Пусть задана функция $f : S \to [0, 1]$, $\forall \omega \in S \exists! f(\omega) \in [0, 1]$.
$\sum\limits_{\omega \in S} f(\omega) = 1$
Определим $\forall A \subseteq S\ Pr(A) = \sum\limits_{\omega \in A} f(\omega)$, в частности:
$Pr(\varnothing) = 0$
$Pr(S) = 1$
$Pr(\{\omega\}) = f(\omega)$
В этот момент получается, что исходная функция f нам как бы уже не нужна, нам достаточно иметь Pr.
(S, Pr) собственно и называется вероятностным пространством.
S называют пространством элементарных событий,
$\omega \in S$ - элементарным событием (исходом),
$A \subseteq S$ - событием
Pr(A) - вероятностью A
$A, B \subseteq S, Pr(A \cap B) = 0$ - несовместными событиями.
$Pr\{P(x)\}$ - таким образом мы будем обозначать вероятность $P(\{\omega \in S\ :\ P(\omega)\})$ множества таких элементарных исходов в S, что для них выполняется условие P. Причем условие мы можем записывать в вольном формате, например $Pr\{$Сборная России по футболу выиграет чемпионат мира$\}$.

5.1.2 Свойства вероятности

- $\forall A, B \subseteq S\ Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$ (формула включений и исключений).
  Доказательство: $Pr(A \cup B) = \sum\limits_{\omega \in A \cup B} f(\omega) = \sum\limits_{\omega \in A} f(\omega) + \sum\limits_{\omega \in B} f(\omega) - \sum\limits_{\omega \in A \cap B} f(\omega)$

- $\forall A \subseteq S\ \overline{A} = S \setminus A$. $Pr(A) + Pr(\overline{A}) = 1$. Доказывается из определения вероятности.

- $Pr(A \cup B) \leq Pr(A) + Pr(B)$. Очевидно из пункта 1 и того факта, что вероятность неотрицательна.

- $Pr(A) = Pr(A\ B) + Pr(A \cap B)$

5.1.3 Парадокс Монти Холла

Аааааааавтомобиииииль! (С)
На некотором телешоу ведущий предлагает игроку выбрать одну из трех дверей. Известно, что за одной из дверей находится автомобиль, а за двумя другими - по козе. После того как игрок сделал выбор, ведущий открывает одну из двух оставшихся дверей (причем обязательно ту, за которой коза, открыть дверь с автомобилем он не может) и предлагает игроку изменить выбор. Вопрос, собственно, в том, стоит ли менять выбор?

Для начала формализуем задачу:

- Нет оснований полагать, что приз скорее за одной дверью, чем за другой (организаторы выбирали дверь наугад)
- Нет оснований полагать, что игрок предпочитает одну дверь другой (игрок выбирает дверь наугад)
- Нет оснований полагать, что если у ведущего есть выбор, он предпочтет одну дверь другой
- Нет оснований полагать, что кто-то из участников процесса нарушает правила игры
  0

Исходя из этого, построим дерево вариантов:
Итак, сначала организаторы случайно (то есть, вероятность каждого из трех выборов - $\frac{1}{3}$) выбирают дверь, за которой помещают автомобиль (A1, A2, A3). Затем игрок делает свой выбор (P1, P2, P3), тоже случайно. Затем ведущий выбирает, какую дверь ему открыть. Заметим, что выбор у ведущего есть только если игрок исходно выбрал дверь, за которой находится приз. В таком случае, мы считаем, что ведущий делает выбор случайным образом (то есть, вероятность каждого из 2 выборов - $\frac{1}{2}$). Таким образом, мы получили набор элементарных исходов (листья дерева), каждый из которых имеет вид (Ax, Py, Hz) - выбор организаторов, выбор игрока, выбор ведущего. Вероятность каждого из таких исходов можно посчитать как произведение вероятностей на пути из корня дерева в лист, соответствующий данному исходу (то есть, $Pr(\{(A1, P1, H2)\}) = \frac{1}{3} * \frac{1}{3} * \frac{1}{2} = \frac{1}{18}$). Теперь нам нужно посчитать $Pr\{$Игрок выиграет, если сменит выбор$\}$. На картинке все исходы, удовлетворяющие этому условию, отмечены буквой W. Посчитав сумму их вероятностей, получим $\frac{2}{3}$.
Почему так получается? Очень грубо и не совсем корректно, зато интуитивно понятно, это можно объяснить так: вероятность того, что игрок исходно угадал - $\frac{1}{3}$. То есть с вероятностью $\frac{2}{3}$ автомобиль за одной из двух других дверей. Когда ведущий открывает дверь, мы не получаем никакой новой информации, т.к. заранее известно, что он должен был открыть дверь с козой. Значит, исходные $\frac{2}{3}$ вероятности, что игрок выбрал не ту дверь, остаются и сосредотачиваются на оставшейся двери, а значит, сменив выбор, игрок с вероятностью $\frac{2}{3}$ выиграет.
Еще до кучи всяких примеров с бросками монетки, бросками кубиков, вытягиванием карт, игрой в русскую рулетку и.т.д можно придумать и самим.

5.1.4 Условная вероятность

Пусть есть вероятностное пространство (S, PR), $A, B \subseteq S$, $Pr(B) \neq 0$. Тогда вероятностью A при условии B (вероятность события A при условии, что известно, что событие B произошло) называют
$Pr(A|B) = \frac{Pr(A \cap B)}{PrB)}$.

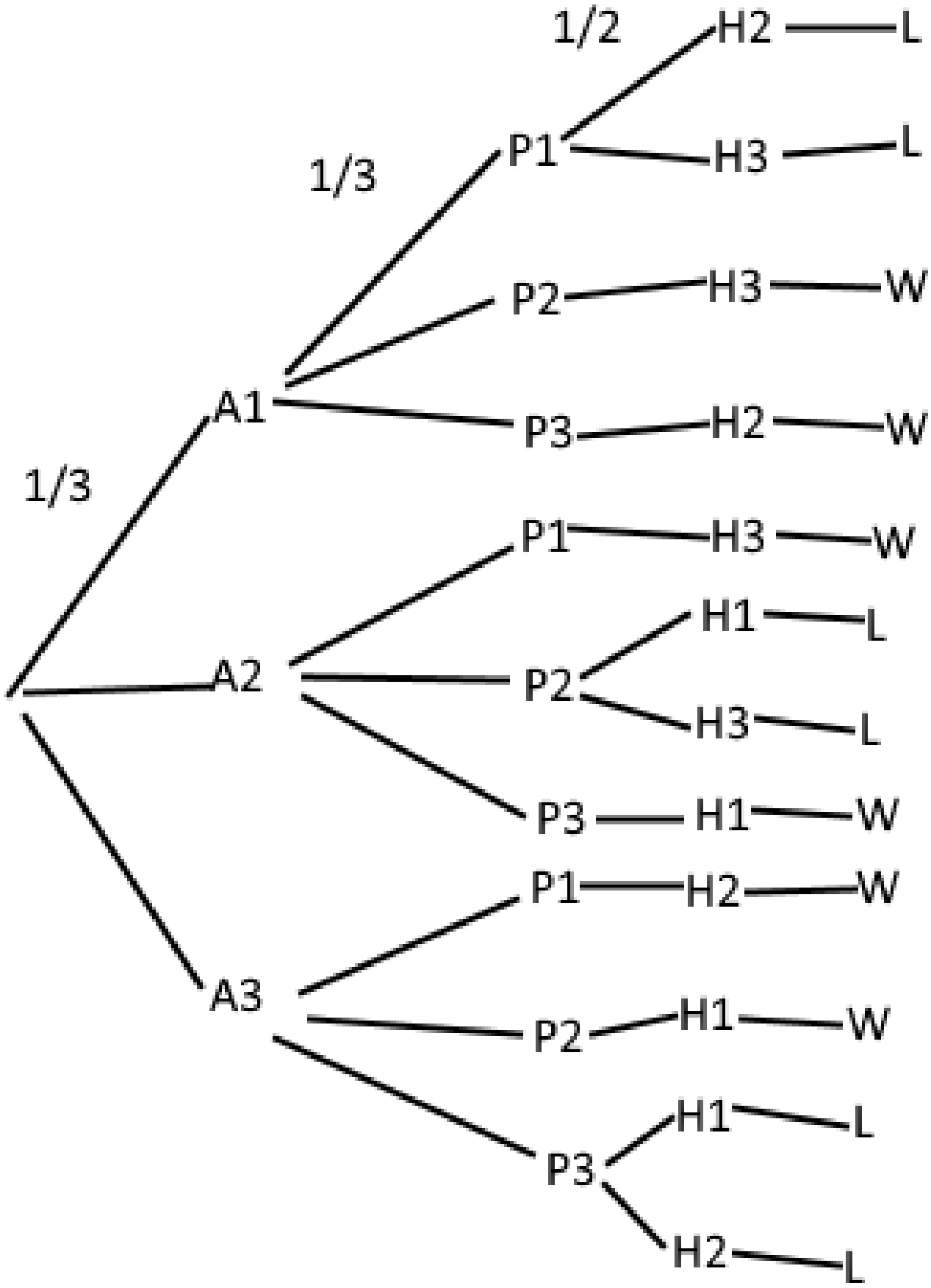

Рис. 1: Дерево вариантов

Рис. 2: Дерево вариантов

5.1.5 Задача о хоккейной команде

ХК Локомотив играет серию до двух побед против СКА. Вероятность того, что Локомотив выиграет первую игру - $\frac{1}{2}$, а вот для остальных игр действует правило: если Локомотив выиграл предыдущую игру, вероятность его победы поднимается до $\frac{2}{3}$, а если проиграл - падает до $\frac{1}{4}$. Построим дерево возможных вариантов:
Построим множество элементарных исходов $S = \{WW, WLW, WLL, LWW, LWL, LL\}$
Посчитаем несколько вероятностей:

- $Pr\{$Локомотив выиграет серию$\} = Pr(\{WW, WLW, LWW\}) = \frac{1}{3} + \frac{1}{24} + \frac{1}{12} = \frac{11}{24}$
- $Pr\{$Локомотив выиграет серию, если он выиграл первую игру$\} = Pr(\{WW, WLW, LWW\}|\{WW, WLW, WLL\}) = \frac{Pr(\{WW,WLW\})}{Pr(\{WW,WLW,WLL\})} = \frac{\frac{1}{3}+\frac{1}{24}}{\frac{1}{2}} = \frac{3}{4}$
- $Pr\{$Локомотив выиграл первую игру, если он выиграл серию$\} = Pr(\{WW, WLW, WLL\}|\{WW, WLW, LWW\}) = \frac{Pr(\{WW,WLW\})}{Pr(\{WW,WLW,LWW\})} = \frac{\frac{1}{3}+\frac{1}{24}}{\frac{1}{2}} = \frac{9}{11}$
- $Pr\{$Локомотив выиграет вторую игру, если он выиграл первую игру$\} =$

$$Pr(\{WW, LWW, LWW\}|\{WW, WLW, WLL\}) = \frac{Pr(\{WW\})}{Pr(\{WW,WLW,WLL\})} = \frac{\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$$

5.2 16.11.18

(Большая часть этой лекции состояла из работы с задачей про хоккейную команду, что я включил в конспект прошлой лекции и из примера про вытаскивание шариков из урн, который мне разбирать лень. Так что всего один пункт. Всегда бы так, а.

5.2.1 Свойства условной вероятности, формула Байеса

- $Pr((A \cap B)|C) = Pr(A|(B \cap C)) * Pr(B|C)$. Доказывается проверкой по формуле условной вероятности.

- Формула Байеса: $Pr(B|A) * Pr(A) = Pr(A \cap B) = Pr(A|B) * Pr(B)$. Доказывается проверкой по формуле условной вероятности.

- $Pr(A_1 \cap \ ... \ \cap A_n) = Pr(A_1|(A_2 \cap \ ... \ \cap A_n)) * \ ... \ * Pr(A_{n-1}|A_n) * Pr(A_n)$. Доказывается по индукции.

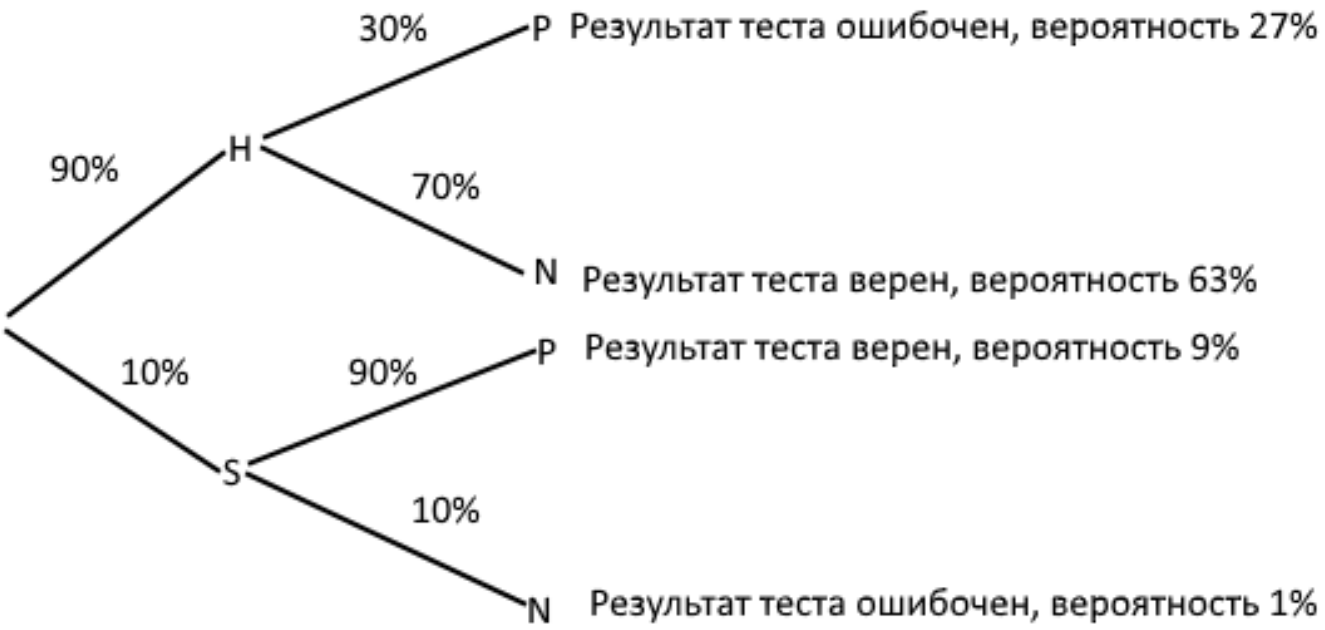

Рис. 3: Дерево вариантов

5.3 23.11.18

5.3.1 Задача о медицинском обследовании

Пусть существует некая болезнь "СПИД ноги которой подвержены 10% населения.
Есть тест на выявление этой болезни, который:

- Имеет 10% вероятность ложного отрицания (вероятность отрицательного (N) результата теста, если человек болен (S)).

- Имеет 30% вероятность ложного подтверждения (вероятность положительного (P) результата теста, если человек здоров (H)).

Построим дерево вариантов:
Посчитаем несколько вероятностей:

- $Pr\{\text{Тест положительный}\} = Pr(\{HP, SP\}) = 9\% + 27\% = 36\%$. то есть, несмотря на то, что болеет всего 10% населения, тест даст положительный результат более чем в $\frac{1}{3}$ случаев.

- $Pr\{\text{Человек болен, если тест положительный}\} = Pr(\{SP, SN\}|\{HP, SP\}) = \frac{Pr(\{SP\})}{Pr(\{HP,SP\})} = \frac{9\%}{36\%} = 25\%$

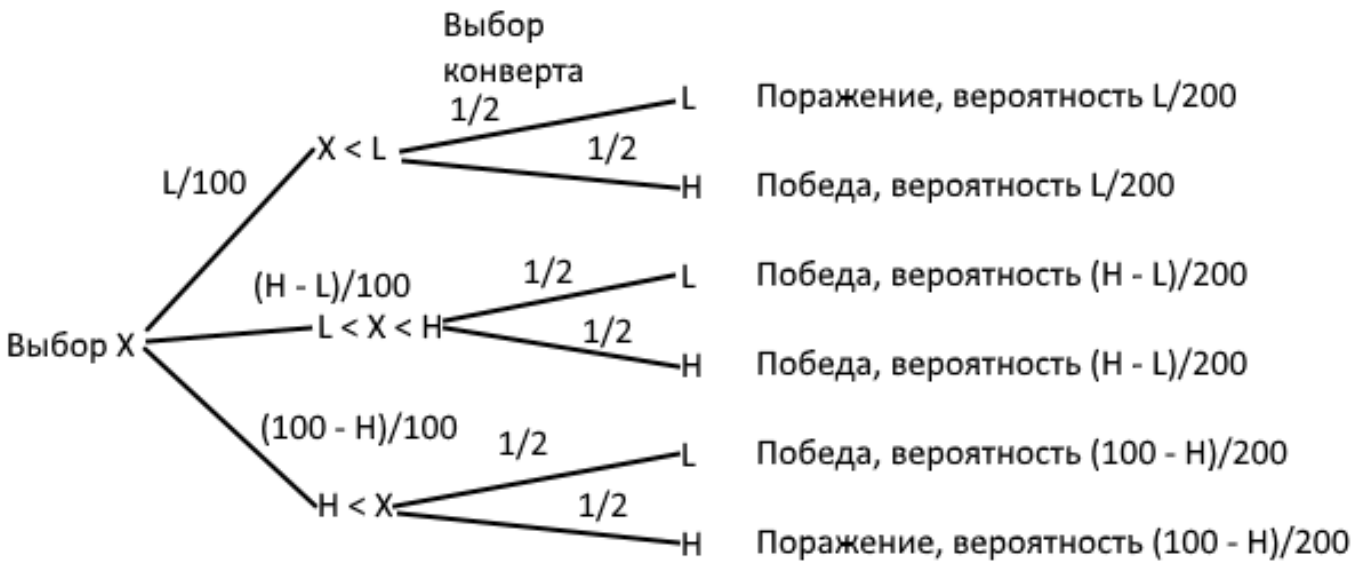

Рис. 4: Дерево вариантов

- $Pr\{\text{Тест работает верно}\} = Pr(\{HN, SP\}) = 63\% + 9\% = 72\%$

Кажется, что тест не очень эффективен. Но так ли это?
Рассмотрим инновационный тест (спонсированный военкоматом), который всегда выдает отрицательный результат.
Для такого чудесного теста $Pr\{\text{Тест работает верно}\} = Pr(\{HN, SP\}) = 90\% + 0\% = 90\%$.
Мораль: нужно аккуратно работать с вероятностями и не поддаваться на провокации.

5.3.2 Задача об угадывании чисел в конвертах

Итак, пусть есть два различных числа от 0 до 100 в закрытых конвертах. Вам предложено, открыв один из конвертов (наугад, то есть, конверты ничем не отличаются), попытаться угадать, открыли вы конверт с большим из двух чисел (H) или меньшим (L). Задача в том, чтобы научиться угадывать с вероятностью хотя бы чуть-чуть лучшей, чем 50% при ЛЮБОЙ стратегии противника.
Давайте попробуем найти такое число X, что $L < X < H$. По имеющемуся X и числу из открытого конверта можно очень легко понять, какое из двух чисел меньше (если число из открытого конверта меньше X, то это L, иначе - H).
Чтобы избежать ситуаций, когда X совпадает с L или H, будем выбирать X из множества $A = \{\frac{1}{2},\ \dots, 99\frac{1}{2}\}$. Вопрос, собственно, в том, как выбрать хорошее число X?
Давайте попробуем просто наугад! Нарисуем дерево вариантов:
Итак, посчитаем вероятность победы с использованием данной стратегии:
$Pr\{\text{победить}\} = \frac{L}{200} + \frac{H-L}{200} + \frac{H-L}{200} + \frac{100-H}{200} = \frac{100-H+H-L+L}{200} + \frac{H-L}{200} = \frac{1}{2} + \frac{H-L}{200}$,
то есть даже в худшем случае, когда противник выбрал два соседних числа, вероятность победить, используя данную стратегию составляет $50\frac{1}{2}\%$.

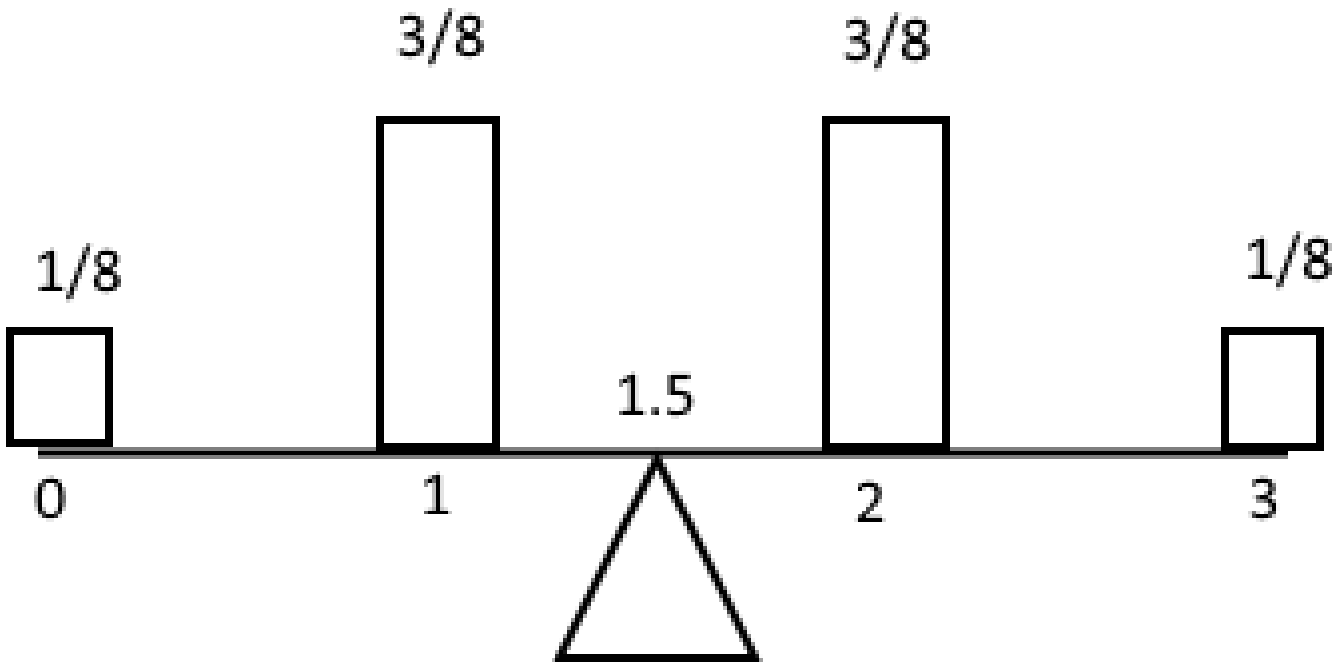

Рис. 5: Матожидание количества орлов

5.3.3 Дискретная случайная величина

Для вероятностного пространства (S, Pr), $|S| < \infty$ функция $\xi : S \to \mathbb{R}$ называется дискретной случайной величиной (далее - ДСВ).
$Pr\{\xi = a\} = Pr(\{\omega \in S \ : \ \xi(\omega) = a\})$
$Pr\{\xi \leq a\} = Pr(\{\omega \in S \ : \ \xi(\omega) leq a\})$
Например для вероятностного пространства, иллюстрирующего три броска монетки, можно ввести дискретную случайную величину ϕ, отражающую количество выпавших орлов. Тогда, например, $Pr\{\phi = 2\} = Pr(\{\text{POO,OPO,OOP}\}) = \frac{3}{8}$

5.3.4 Математическое ожидание ДСВ

Математическим ожиданием (матожиданием) ДСВ ξ на вероятностном пространстве (S, Pr) называют
$E\xi = \sum\limits_{\omega \in S} \xi(\omega) * Pr(\{\omega\})$
Альтернативная формула матожидания:
$E\xi = \sum\limits_{a \in Im(\xi)} \sum_{\omega \in S, \xi(\omega) = a} a * Pr(\{\omega\}) = \sum\limits_{a \in Im(\xi)} a * \sum_{\omega \in S, \xi(\omega) = a} Pr(\{\omega\}) = \sum\limits_{a \in Im(\xi)} a * Pr(\{\omega \in S \ : \ \xi(\omega) = a\}) = \sum\limits_{a \in Im(\xi)} a * Pr\{\xi = a\}$
Также можно заметить, что если изобразить числовую прямую как балку, и на каждой точке a из $Im(\xi)$ нарисовать столбик высоты $Pr\{\xi = a\}$, матожидание ξ будет лежать на числовой прямой в той точке, на которой эту балку можно сбалансировать.
Вот рисунок для матожидания количества орлов, выпавших после броска трех монет:

5.3.5 Арифметические действия с ДСВ и матожиданием

- $\eta = \xi + c$ определим как $\eta(\omega) = \xi(\omega) + c$. $Pr\{\eta = a\} = Pr\{\xi + c = a\} = Pr\{\xi = a - c\}$. По второй формуле матожидания можно заметить, что $E\eta = c + E\xi$

- $\eta = \xi * c$, $c \neq 0$ определим как $\eta(\omega) = \xi(\omega) * c$. $Pr\{\eta = a\} = Pr\{\xi * c = a\} = Pr\{\xi = \frac{a}{c}\}$. По второй формуле матожидания можно заметить, что $E\eta = c * E\xi$

- $\eta = \xi^2$ определим как $\eta(\omega) = (\xi(\omega))^2$. $Pr\{\eta = a\} = Pr\{\xi^2 = a\}$. $E\eta = \sum\limits_{\omega \in S} \xi^2(\omega) * Pr(\{\omega\})$

- $\eta = \xi + \psi$ определим как $(\xi + \psi)(\omega) = \xi(\omega) + \psi(\omega)$. $E($
  $xi + \psi) = \sum\limits_{\omega \in S} (\xi(\omega) + \psi(\omega)) * Pr(\{\omega\}) = \sum\limits_{\omega \in S} \xi(\omega) * Pr(\{\omega\}) + \sum\limits_{\omega \in S} \psi(\omega) * Pr(\{\omega\}) = E\xi + E\psi$

5.3.6 Дисперсия ДСВ

Дисперсией ДСВ называют матожидание квадрата разности значения ДСВ и ее матожидания: $D\xi = E(\xi - E\xi)^2$.
$E(\xi - E\xi)^2 = E(\xi^2 - 2 * E\xi * \xi + (E\xi)^2) = E\xi^2 - E(2 * E\xi * \xi)) + E(E\xi)^2 = E\xi^2 - 2 * E\xi * E\xi + (E\xi)^2 = E\xi^2 - (E\xi)^2$
Таким образом, $D\xi = E\xi^2 - (E\xi)^2$.

5.4 30.11.18

5.4.1 Испытание Бернулли

Пусть на произвольном вероятностном пространстве (S, Pr) объявлена ДСВ ξ такая, что $Pr\{\xi = 1\} = p, Pr\{\xi = 0\} = 1 - p$. Словами это можно обосновать так: каждый элементарный исход $\omega \in S$ считается либо успешным ($\xi(\omega) = 1$), либо неудачным ($\xi(\omega) = 0$). Тогда:
$E\xi = 0 * (1 - p) + 1 * p = p$
$D\xi = E\xi^2 - (E\xi)^2 = (1 - p) * 0^2 + p * 1^2 - p^2 = p - p^2 = p(1 - p)$

5.4.2 Моделирование ДСВ, генераторы случайных чисел

Многократно повторяемый эксперимент, исходы которого имеют заданный набор вероятностей, соответствующий набору вероятностей ДСВ, называют моделью этой ДСВ.
Пример: моделью ξ ($Pr\{\xi = 0\} = \frac{1}{2}, Pr\{\xi = 1\} = \frac{1}{2}$) можно считать бросок монетки (если считать, что стороны монетки одинаковые, монетка никогда не падает на ребро и бросают ее наугад).
Многократно повторяемый эксперимент, результатом которого является (псевдо)случайно выбранное число из некоторого конечного множества, называют дискретным генератором (датчиком) случайных чисел.
Пример: назовем α дискретный генератор случайных чисел, для которого $Pr\{\alpha = k\} = \frac{1}{N}\ \forall k \in 1 : N$.
Многократно повторяемый эксперимент, результатом которого является (псевдо)случайно выбранное число из некоторого промежутка числовой прямой, называют непрерывным генератором (датчиком) случайных чисел.
Пример: назовем α_0 непрерывный генератор случайных чисел, который с равной вероятностью попадает в каждую из точек отрезка $[0, 1]$.
Поскольку отрезок $[0, 1]$ содержит несчетное количество точек, вероятность $Pr\{\alpha_0 = x\} = \frac{1}{\infty} = 0\ \forall x \in [0, 1]$. Однако, вероятность того, что точка попадет в некоторый промежуток, все же ненулевая:
$Pr\{x \in < a, b >\} = \frac{|<a,a+\delta>|}{|[0,1]|} = \frac{\delta}{1} = \delta$
В частности, $Pr\{x \in [0, 1]\} = 1$.
Замечание: поскольку для конкретной точки вероятность попадания туда α_0 равна нулю, можно смело выкидывать из множества конечное (и даже счетное) количество точек, и вероятность попадания в него останется неизменной. Следовательно, вместо $< a, b >$ можно рассматривать (a, b) или $[a, b)$. Последнее особенно удобно, поскольку полуинтервалы хорошо стыкуются.

5.4.3 Табличный метод моделирования ДСВ

Итак, пусть есть вероятностное пространство (S, Pr) и заданная на нем ДСВ ξ.
При этом $S = \{0, 1, \ ... \ , n - 1\}$, $\forall i \in 0 : (n - 1)\ Pr(\{i\}) = p_i, \sum\limits_{i=0}^{n-1} p_i = 1$.
Примечание: для ДСВ с произвольным набором значений можно их пронумеровать и свести задачу к данной. Возьмем отрезок $[0, 1)$ и поделим его на полуинтервалы вида $[x_i, x_{i+1})$, где $x_0 = 0$, $\forall i \in 0 : (n - 1)\ x_{i+1} = x_i + p_i$
Для такого разбиения отрезка $[0, 1]$ вероятность $Pr\{\alpha_0 \in [x_i, x_{i+1}]\} = p_i$. То есть, эксперимент "в отрезок с каким номером попадет случайное число, выданное α_0?"моделирует ξ.

Замечание: При попадании в точку 1 значение α_0 оказывается вне пределов всех полуинтервалов. Но с одной стороны, вероятность такого события - 0, так что распределение вероятностей это нам не портит, а с другой, если это все же произошло, мы можем просто объявить эксперимент неудачным и провести его заново.
Основным недостатком данного метода является большое количество сравнений вещественных чисел, каковая операция является не очень точной и достаточно долгой. Неплохо бы придумать метод, который минимизирует количество таких операций.

5.4.4 Метод Уокера

В табличном методе мы располагали полуинтервалы на отрезке $[0, 1]$ в произвольном порядке. В этот раз мы сначала разделим его на полуинтервалы $[\frac{i}{n}, \frac{i+1}{n}) \ \forall i \in 0 : (n-1)$ (назовем их базовыми полуинтервалами. Такого обозначения на лекции не было, но я не хочу каждый раз уточнять, имею я в виду эти полуинтервалы или те, длины которых равны вероятностям) и будем следить, чтобы в одном таком полуинтервале было не более одной границы полуинтервалов, соответствующих исходам. Более того, если раньше каждому исходу соответствовал ровно один полуинтервал, то сейчас их может быть несколько, главное только, чтобы сумма их длин все еще была равна вероятности этого исхода.
Зачем? Давайте посмотрим на число $n * \alpha_0$.
Утверждение: его целая часть $\lfloor n * \alpha_0 \rfloor$ будет равно i, если $\frac{i}{n} \leq \alpha_0 < \frac{i+1}{n}$.
Доказательство: умножим все части неравенства на n: $i \leq n * \alpha_0 < i + 1 \Leftrightarrow \lfloor n * \alpha_0 \rfloor = i$.
Итак, мы научились понимать, в какой базовый полуинтервал попал α_0. Теперь осталось понять, что делать, если в базовом полуинтервале есть граница между полуинтервалами. Пусть эта граница есть в точке $x, \frac{i}{n} \leq x < \frac{i+1}{n}$. Растянем наш базовый полуинтервал в n раз:
Точка $\frac{i}{n}$ перейдет в точку $i = \lfloor n * \alpha_0 \rfloor$, α_0 - в $n * \alpha_0$, x - в $n * x$, $\frac{i+1}{n}$ - в $i + 1 = \lfloor n * \alpha_0 \rfloor + 1$.
Теперь сдвинем полуинтервал влево на i: $\lfloor n * \alpha_0 \rfloor$ перейдет в 0, $n * \alpha_0$ - в $\{n * \alpha_0\}$ (дробная часть), $n * x$ - в $n * x - i$, $\lfloor n * \alpha_0 \rfloor + 1$ - в 1. То есть, теперь нам достаточно сравнить $\{n * \alpha_0\}$ и $n * x - i$, причем $n * x - i$ не зависит от α_0 и может быть посчитана заранее.
Замечание: при попадании в единицу нам все еще придется повторять опыт.
Итак, пусть каждому значению случайной величины $i \in 0 : (n-1)$ соответствует множество P_i полуинтервалов, причем каждый из них полностью лежит в одном из базовых полуинтервалов и пересечение P_i и P_j пусто для любого $j \neq i$, а $\sum\limits_{[a,b) \in P_i} |[a, b)| = p_i$. Также, в каждом базовом полуинтервале лежит не более двух полуинтервалов.
Тогда мы научились с помощью двух сравнений понимать, в какой полуинтервал $[a, b)$ попал α_0, а если объявить исходом эксперимента i, где $[a, b) \in P_i$, то этот эксперимент будет моделировать ξ, так как вероятность попасть в полуинтервал, лежащий в P_i равна $\sum\limits_{[a,b) \in P_i} Pr\{\alpha_0 \in [a, b)\} = \sum\limits_{[a,b) \in P_i} |[a, b)| = p_i$ Осталось только построить такое разбиение.
Для этого нам придется доказать две леммы:

Лемма 1: если $n > 1$, $\exists l \in 0 : (n-1) : \ p_l \leq \frac{1}{n}$. Доказательство: если нет, то $\sum\limits_{i=0}^{n-1} p_i > 1$.

Лемма 2: если $n > 1$, $\forall l \in 0 : (n-1) \ \exists m \in 0 : (n-1), m \neq l : \ p_l + p_m > \frac{1}{n}$. Доказательство: если нет, то $\exists l_0 \in 0 : (n-1)$ такое, что:
$\sum\limits_{m \neq l_0} (p_{l_0} + p_m) \leq \frac{n-1}{n} < 1$
С другой стороны, $\sum\limits_{m \neq l_0} (p_{l_0} + p_m) = \sum\limits_{m \neq l_0} p_{l_0} + \sum\limits_{m \neq l_0} p_m = (n-1) * p_{l_0} + \sum\limits_{m \in 0:(n-1)} p_m \geq 1$. Противо-

речие.
Итак, начнем строить наше разбиение:
Определим $\xi^{(0)} : p_i^{(0)} = p_i$, выберем $l_0 : \ p_{l_0}^{(0)} < \frac{1}{n}$, $m_0 : \ p_{l_0}^{(0)} + p_{m_0}^{(0)} > \frac{1}{n}$
Определим ДСВ $\psi^{(0)} : \ Pr\{\psi^{(0)} = l_0\} = n * p_{l_0}^{(0)}\}, Pr\{\psi^{(0)} = m_0\} = 1 - n * p_{l_0}^{(0)}\}$.
Теперь отрежем от отрезка $[0, 1]$ первый базовый полуинтервал:
$\xi^{(1)} : p_{l_0}^{(1)} = 0, p_{m_0}^{(1)} = (p_{m_0}^{(0)} - (\frac{1}{n} - p_{l_0}^{(0)})) * \frac{n}{n-1}, p_i^{(1)} = p_i * \frac{n}{n-1}$ и перейдем к следующему шагу алгоритма.
Теперь научимся вычислять $\forall k \in 0 : (n-1) \ p_i^{(0)}$ через $\xi^{(1)}$:
$p_k^{(0)} = \frac{1}{n} Pr\{\psi^{(0)} = k\} + \frac{n-1}{n} p_k^{(1)}$
Корректность этой формулы можно аккуратно руками проверить.
Теперь рассмотрим общий случай:
Пусть построено $\xi^{(i)}$, $i \geq 1$, у этой ДСВ $n - i$ возможных исходов.
$\exists l_i, m_i : \ m_i \neq l_i, p_{l_i}^{(i)} \leq \frac{1}{n-i}, p_{m_i}^{(i)} + p_{l_i}^{(i)} > \frac{1}{n-i}$.
$\psi^{(i)} : \ Pr\{\psi^{(i)} = l_i\} = (n-i) * p_{l_0}^{(i)}, Pr\{\psi^{(i)} = m_i\} = 1 - (n-i) * p_{l_0}^{(i)}$
Строим $\xi^{(i+1)}$: $p_{l_i}^{(i+1)} = 0, p_{m_i}^{(i+1)} = p_{m_i}^{(i)} - \frac{n-i}{n-i-1}(\frac{1}{n-i} - p_{l_i}^{(i)}), p_s^{(i+1)} = \frac{n-i}{n-i-1} p_s^{(i)}$
Тогда $p_k^{(i)} = \frac{1}{n-i} Pr\{\psi^{(i)} = k\} + \frac{n-i-1}{n-i} p_k^{(i+1)}$.
Так до $n - 1$ шага. На $n - 1$ шаге получаем, что $\xi^{(n-1)}$ имеет всего один возможный исход: $Pr\{\xi^{(n-1)} = l_{n-1}\} = 1$.
Тогда $p_k^{(n-1)} = \frac{1}{1} Pr\{\psi^{(n-1)} = k\} + 0 = Pr\{\psi^{(n-1)} = k\}$.
Таким образом, мы получили рекурсивную формулу для $p_k^{(i)}$. Раскроем ее:
$p_k = p_k^{(0)} = \frac{1}{n} Pr\{\psi^{(0)} = k\} + \frac{n-1}{n} p_k^{(1)} = \frac{1}{n} Pr\{\psi^{(0)} = k\} + \frac{n-1}{n}(\frac{1}{n-1} Pr\{\psi^{(1)} = k\} + \frac{n-2}{n-1} p_k^{(2)}) = \frac{1}{n} Pr\{\psi^{(0)} = k\} + \frac{1}{n} Pr\{\psi^{(1)} = k\} + \frac{n-2}{n} p_k^{(2)} = ... = \frac{1}{n} \sum\limits_{j \in 0:(n-1)} Pr\{\psi^{(j)} = k\}$
. Пример: $p_0 = 0.91, p_1 = p_2 = p_3 = p_4 = p_5 = p_6 = p_7 = p_8 = p_9 = 0.01$.
На первом шаге $l_0 = 1, m_0 = 0$. Тогда $p_1^{(1)} = 0$, $p_0^{(1)} = \frac{10}{9}(p_0^{(0)} - (\frac{1}{10} - p_0^{(0)})) = \frac{10}{9}(p_0 - \frac{1}{10} + p_1) = \frac{10}{9}(0.91 - 0.1 + 0.01) = \frac{10}{9} 0.82$, $p_i^{(1)} = \frac{10}{9}(p_i^{(0)}) = \frac{0.1}{9}$.
И так далее, на последнем шаге получим $p_0^{(9)} = 1$, $p_i^{(9)} = 0$.

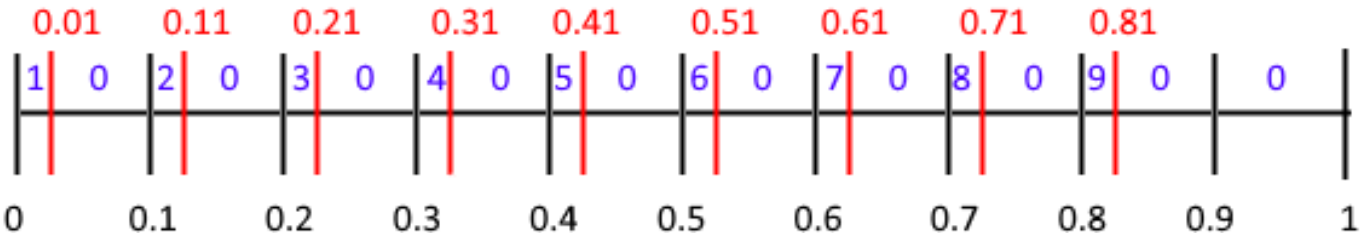

Рис. 6: Разбиение

На рисунке черным обозначены границы базовых полуинтервалов, красным - границы полуинтервалов, синим - номера исходов, соответствующих данным полуинтервалам.

5.5 07.12.18

5.5.1 Вычислительная схема метода Уокера

Заметим, что постоянное домножение на коэффициенты в методе Уокера нужно только чтобы на каждом шаге $\xi^{(i)}$ оставалась ДСВ, что необходимо для формального доказательства, но совершенно не нужно для практических вычислений.
В приведенной ниже таблице коэффициенты вынесены за скобки (самый левый столбец), и по большей части игнорируются.

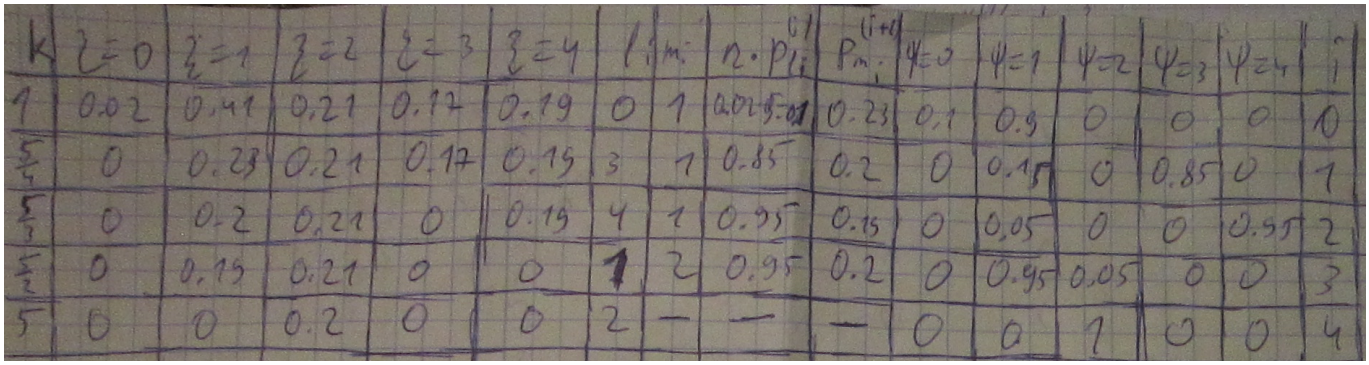

Рис. 7: Вычислительная схема

Следующие 5 столбцов показывают набор вероятностей для $\xi^{(i)}$, затем идут номер l_i, номер m_i, значение $n*p_{l_i}^{(i)}$, необходимое для построения $\psi^{(i)}$, значение $p_{m_i}^{(i+1)} = p_{m_i}^{(i)} - \frac{1}{n} + p_{l_i}^{(i)}$ (без домножения на коэффициент!), необходимое для пересчета $\xi^{(i+1)}$, затем набор вероятностей $\psi^{(i)}$, ну и наконец само i. Здесь стоит заметить, что i - это не только номер текущего шага алгоритма, но и номер базового полуинтервала, который мы на этом шаге алгоритма делим.
Как же пользоваться этой схемой? Рассмотрим на примере:
Пусть $\alpha_0 = 0.73$. Тогда $n * \alpha_0 = 5 * 0.73 = 3.65$. $\lfloor n * \alpha_0 \rfloor = 3$, значит, мы попали в третий базовый полуинтервал. Граница между полуинтервалами в нем проходит по $n * p_{l_i}^{(i)} = 5 * p_{l_3}^{(3)} = 0.95$. Дробная часть $\{n * \alpha_0\} = 0.65 < 0.95$, значит, результат эксперимента - $l_i = l_3 = 1$.

5.5.2 Моделирование ДСВ с помощью последовательности (псевдо)случайных бит.

Псевдослучайный бит - 0 или 1 равновероятно (бросок монетки).
Чего мы хотим? Научиться моделировать ДСВ, вероятности которой - рациональные двоичные числа, с помощью последовательности случайных бит.
Пример: $\xi :\ Pr\{\xi = 0\} = \frac{1}{4} = 0.01_2, Pr\{\xi = 1\} = \frac{1}{2} = 0.1_2, Pr\{\xi = 2\} = \frac{1}{4} = 0.01_2$
Казалось бы, можно построить полное двоичное дерево нужной глубины h (равной максимальному количеству значащих двоичных цифр после запятой среди p_i) и каким-то образом распределить листья по исходам так, чтобы количество листьев, соответствующих i-му исходу было равно $2^h * p_i$. Однако, посмотрим на два различных корректных способа распределения листьев между исходами:

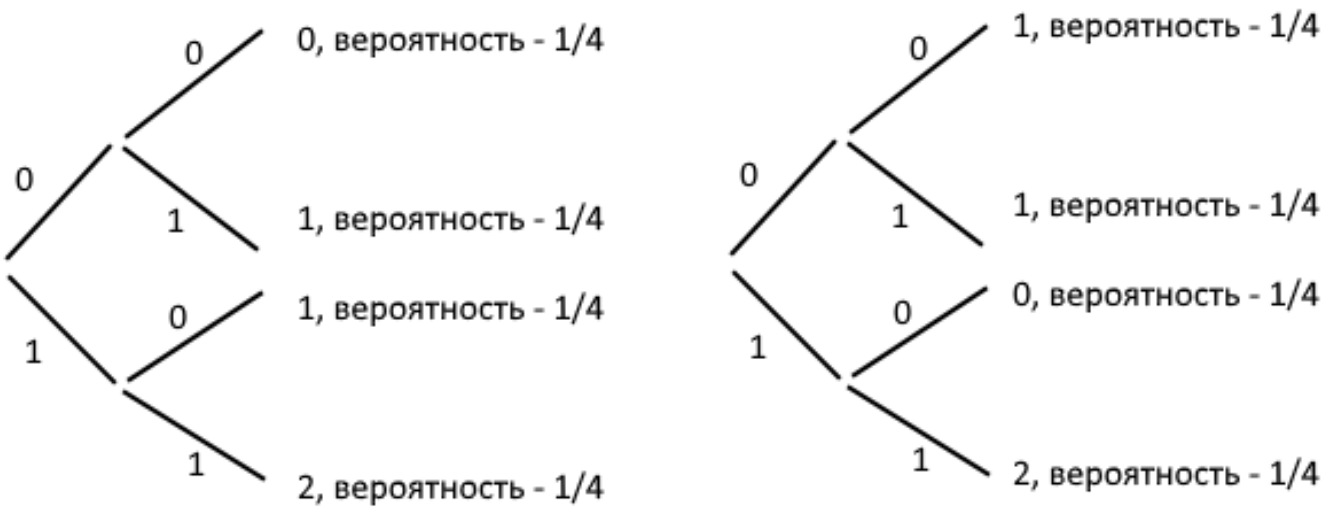

Рис. 8: Сравнение способов

В первом случае нам в любом случае придется бросать монетку дважды, а вот во втором если первый бросок монетки выдал результат 0, дальше можно уже не бросать - нам точно известно, что результатом эксперимента будет 1.
Собственно, задача изложенного ниже алгоритма в том, чтобы минимизировать среднее число бросков монетки, необходимое для получения результата. Такое разбиение множества листьев будем называть оптимальным.
Формализуем:
Пусть есть ДСВ ξ, $\forall i \in 0 : (n-1)\ Pr\{\xi = i\} = p_i$, $\sum\limits_{i \in 0:(n-1)} p_i = 1$.

p_i в двоичной записи имеет представление $0.\pi_1^i\pi_2^i...\pi_m^i$, то есть $p_i = \frac{\pi_1^i}{2} + \frac{\pi_2^i}{4} + ... + \frac{\pi_m^i}{2^m}$, где $\pi_k^i \in \{0, 1\}$. Тогда $2^m p_i = 2^{m-1}\pi_1^i + ... + \pi_m^i$.
Рассмотрим множество A исходов m бросков монетки. $|A| = 2^m$.
Рассмотрим разбиение $A = A_0 \cup A_1 \cup ... \cup A_{n-1}$, где $\forall i \in 0 : (n-1)\ |A_i| = 2^m p_i$.
Тогда если исход эксперимента "m бросков монетки"лежит в A_i, то регистрируем исход i. $Pr\{$зарегистрирован исход i$\} = \frac{|A_i|}{2^m} = p_i$. Но как построить оптимальное разбиение?
Заведем набор множеств $\forall k \in 1 : m\ I_k = \{i \in 0 : (n-1)\ :\ \pi_k^i = 1\}$.
Пусть мы бросили монету 1 раз. Множество исходов B_1 имеет мощность 2. Выберем какое-то $M_1 \subseteq B_1$ такое, что $|M_1| = |I_1| = \sum\limits_{j \in 0:(n-1)} \pi_1^j$. Установим биекцию между M_1 и I_1. Во всех этих случаях исход уже определен, а в остальных $B_1 \setminus M_1$ случаях нам нужно продолжать бросать. Теперь у нас есть $B_2, |B_2| = 2|B_1 \setminus M_1|$, продолжаем использовать тот же алгоритм.
В общем случае, у нас есть B_k - множество исходов после k бросков монетки, выбираем из них множество $M_k \subseteq B_k$ исходов такое, что $|M_k| = |I_k|$, строим биекцию между M_k и I_k, определяя исход эксперимента для этих случаев, а в остальных $B_k \setminus M_k$ случаях продолжаем бросать.
Пример:
$p_0 = 0.101001; p_1 = 0.000001; p_2 = 001101; p_3 = 001001$. Тогда:
$I_0 = \{0\}$
$I_1 = \varnothing$
$I_2 = \{0, 2, 3\}$

$I_3 = \{2\}$
$I_4 = \varnothing$
$I_5 = \{0, 1, 2, 3\}$

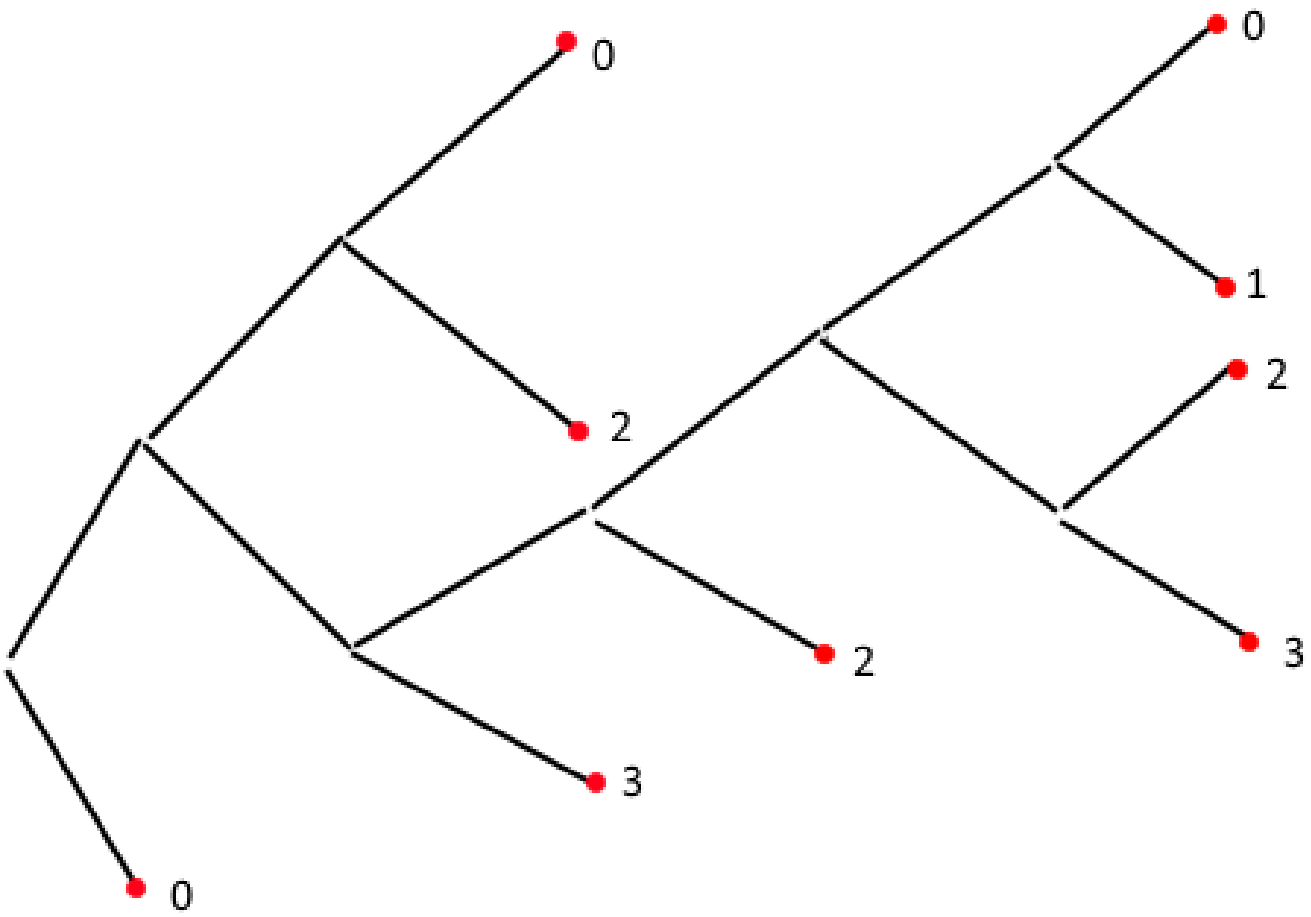

Рис. 9: Дерево вариантов

Таким образом, матожидание числа бросков равно:
$E\{\text{число бросков}\} = \sum\limits_{k \in 1:m} k * Pr\{\text{монета брошена k раз}\} = \sum\limits_{k \in 1:m} k|I_k|\frac{1}{2^k} = \sum\limits_{k \in 1:m} \frac{k}{2^k} \sum\limits_{j \in 0:(n-1)} \pi_k^j.$

5.6 08.12.18

5.6.1 Префиксный код

Λ - произвольное конечное множество (алфавит). $a \in \Lambda$ - символы.
$\forall a \in \Lambda\ \exists l(a) \in \mathbb{N}, \exists c(a) = \{0,1\}^{l(a)}$ - кодовая последовательность a.
$\forall a, b \in \Lambda, a \neq b\ c(a) \neq c(b)$. Казалось бы, это достаточное условие, чтобы по коду однозначно распознавался символ. Но рассмотрим случай:
$\Lambda = \{a, b\}$, $c(a) = 10$, $c(b) = 100$. Тогда последовательность 100 получатель сообщения может начать распознавать как "a??? a поскольку 0 не является валидным кодом, получатель не сможет расшифровать сообщение без каких-то дополнительных усилий. Если же добавить в алфавит символы $d, c(d) = 01$ и $e, c(e) = 1$, сообщение 1001 можно будет понять и как a, и как be.
Чтобы избежать подобных проблем, вводят условие префиксности:
Код называется префиксным, если $\forall a, b \in \Lambda\ c(a) = w \Rightarrow \nexists m \in \mathbb{N}_0 :\ c(b) = w\gamma$, где $\gamma \in \{0,1\}^m$.
Иначе говоря, код называется префиксным, если никакая кодовая последовательность одного символа не является началом кодовой последовательности другого символа. Заметим также, что данное условие включает в себя условие несовпадения кодов различных символов, так что в дальнейшем отдельно рассматривать мы его не будем.

5.6.2 Задача об оптимальном префиксном коде

Пусть каждому символу $a \in \Lambda$ соответствует $p(a)$ - вероятность появления этого символа в сообщении. (Поскольку сообщение состоит из одного символа, $\sum\limits_{a \in \Lambda} p(a) = 1$. Также считаем, что $\forall a \in \Lambda\ p(a) > 0$).
Введем ДСВ $l :\ \forall a \in \Lambda\ Pr\{l = l(a)\} = p(a)$ - длина кодовой последовательности символа в сообщении.
Оптимальным называется префиксный код, минимизирующий матожидание l: $El = \sum\limits_{a \in \Lambda} l(a)p(a)$
Интуитивно понятно, что чем чаще встречается символ, тем короче должна быть его кодовая последовательность. Но вот как это формализовать?
Договоримся про обозначения: здесь и далее считаем, что символы с одинаковыми прибамбасами (волна, палочка, штрих, два штриха и.т.д) относятся к коду с таким же прибамбасом, обговаривать это каждый раз не будем. То есть, $C' = \{(a, l'(a), c'(a))\ :\ a \in \Lambda\}$, например.
Существование оптимального префиксного кода: ну, мы знаем, что $E\,l \geq 1$, ведь в каждой кодовой последовательности должен быть хотя бы один символ. Ну и мы знаем, что в худшем случае мы можем сделать префиксный код, в котором все символы имеют одинаковые длины кодовых последовательностей и эти последовательности различны. В таком случае $\forall a \in \Lambda\ l(a) = \lceil \log_2(|\Lambda|) \rceil$, так что префиксные коды существуют и матожидание длины кодовой последовательности ограничено.
Лемма 1: Если в некотором (не обязательно оптимальном) коде C существует $x \in \Lambda :\ c(x) = w\alpha$, где $\alpha \in \{0,1\}$, и при этом $\nexists y \in \Lambda, y \neq x :\ c(y) = w\gamma$, где $\gamma \in \{0,1\}^k$ (то есть, если w не является началом никакой другой кодовой последовательности, кроме $c(x)$), то код C' такой, что $c'(x) = w, \forall y \in \Lambda, y \neq x\ c'(y) = c(y)$ во-первых будет префиксным (по построению, префиксности C и условию леммы), а во-вторых, $E\,l' = E\,l - p(x)l(x) + p(x)(l(x) - 1) = E\,l - p(x) < E\,l$.
Тогда получается, что код C точно не мог быть оптимальным.

Лемма (*), которой не было в лекции, но она удобная: если в префиксном коде C $\exists a, b \in \Lambda, a \neq b: \ p(a) < p(b), l(a) < l(b)$, то такой код не оптимален. Доказательство: проверим, что для кода C', в котором $c'(a) = c(b), c'(b) = c(a), \forall x \in \Lambda, x \neq a, x \neq b \ c'(x) = c(x)$
$E\ l - E\ l' = p(a)l(a) + p(b)l(b) - p(a)l(b) - p(b)l(a) = p(a)(l(a) - l(b)) - p(b)(l(a) - l(b)) = (p(a) - p(b))(l(a) - l(b)) > 0$.
Лемма 2: Пусть $a, b \in \Lambda, a \neq b$ - два символа с наименьшими вероятностями (для определенности, $\forall x \in \Lambda \ p(x) \leq p(b) \leq p(a)$). Тогда существует оптимальный префиксный код такой, что в нем $c(a) = w0, c(b) = w1$, где $\exists k \in \mathbb{N}_0: \ w \in \{0,1\}^k$, и это самые длинные кодовые последовательности.
Доказательство: пусть C' - какой-то оптимальный префиксный код. По лемме (*) a и b имеют самые длинные кодовые последовательности в C': $\forall x \in \Lambda, x \neq a, x \neq b \ l'(a) \geq l'(b) \geq l'(x)$. Если $c(a) = w\gamma, w \in \{0,1\}^{l(b)}, \gamma \in \{0,1\}^{l(a)-l(b)}$ и w не является началом никакой кодовой последовательности (а он не является, т.к. все остальные кодовые последовательности не длиннее w, и при этом никакой символ не может иметь кодовую последовательность w, так как тогда нарушалась бы префиксность C'), то можно сократить кодовую последовательность a, создав более оптимальный код, что противоречит оптимальности C'.
Таким образом, мы узнали, что поскольку C' оптимален, $l(a) = l(b)$. Ну а теперь дело техники: пусть $c'(b) = w1$ (если заканчивается на ноль, делаем все аналогично и в конце меняем коды местами), тогда если $\exists x \in \Lambda: \ c'(x) = w0$, построим оптимальный (поскольку длины кодовых последовательностей не изменились) код C: $c(a) = c'(x), c(x) = c'(a), \forall z \in \Lambda, z \neq a, z \neq x \ c(z) = c'(z)$. Если же такого x не нашлось, то построим оптимальный (поскольку длины кодовых последовательностей не изменились) код C: $c(a) = w0, \forall z \in \Lambda, z \neq a \ c(z) = c'(z)$. Лемма доказана.
Лемма 3: $\forall x \in \Lambda, x \neq a, x \neq b \ p(a) \leq p(b) \leq p(x)$.
$\Lambda' = \Lambda \setminus \{a, b\} \cup \{\underbrace{ab}\}$, где $\underbrace{ab} \notin \Lambda, p(\underbrace{ab}) = p(a) + p(b)$.
Пусть C' - оптимальный префиксный код для Λ', $c'(\underbrace{ab}) = w$. Тогда для Λ код C:
$c(a) = w0, c(b) = w1, \forall x \in \Lambda, x \neq a, x \neq b \ c(x) = c'(x)$ будет оптимальным префиксным кодом.
Доказательство: $l(a)p(a) + l(b)p(b) = (l'(\underbrace{ab}) + 1)(p(a) + p(b)) = l'(\underbrace{ab})p(\underbrace{ab}) + p(\underbrace{ab})$ Тогда $E\ l = E\ l' + p(\underbrace{ab})$.
Пусть $\overline{C}$ - оптимальный код для Λ, причем $E\ \overline{l} < E\ l$.
По лемме 2: $\overline{c}(a) = \gamma 0, \overline{c}(b) = \gamma 1$.
Построим $\overline{C}'$ для Λ': $\overline{c}'(\underbrace{ab}) = \gamma, \forall x \in \Lambda, x \neq a, x \neq b \ \overline{c}'(x) = \overline{c}(x)$
Является ли $'$ префиксным кодом? Да. Никакой символ по лемме (*) не мог иметь кодовую последовательность длины $> \overline{l}(a)$. Никакой символ не мог иметь кодовую последовательность w, так как $\overline{C}$ префиксный. А единственные две последовательности длины $\overline{l}(a)$, начинающиеся на w, - это коды символов a и b, которых в Λ' нет.
При этом, $E\ \overline{l} = E\ \overline{l}' + p(\underbrace{ab})$. Но поскольку по предположению $E\ l = E\ l' + p(\underbrace{ab}) > E\ \overline{l} = E\ \overline{l}' + p(\underbrace{ab})$, получаем $E\ l' > E\ \overline{l}'$, что противоречит оптимальности C' на Λ'.
Значит, предположение неверно и $E\ \overline{l} \geq E\ l$, но так как $\overline{C}$ оптимален, $E\ \overline{l} = E\ l$, то есть C оптимален. Лемма доказана.

5.6.3 Алгоритм Хаффмана построения оптимального префиксного кода.

Итак, нам нужно построить оптимальный префиксный код на алфавите Λ, $|\Lambda| = M$.
Возьмем $\Lambda_0 = \Lambda$.

$\forall k \in 0 : (M-3)$ возьмем $a_k, b_k \in \Lambda_k$ такие, что $\forall x \in \Lambda_k, x \neq a_k, x \neq b_k\ p_k(a_k) \leq p_k(b_k) \leq p_k(x)$ и построим $\Lambda_{k+1} = \Lambda_k \setminus \{a_k, b_k\} \cup \{\underbrace{a_k b_k}\}$.

Для $\Lambda_{M-2} = \{a_{M-2}, b_{M-2}\}$ оптимальным, очевидно, будет код C_{M-2} : $c_{M-2}(a_{M-2}) = 0, c_{M-2}(b_{M-2}) = 1$, так как для него $E\, l_{M-2} = 1$.
Теперь пусть для $k \in 1 : (M-2)$ у нас есть оптимальный префиксный код C_k для Λ_k. По лемме 3 построим оптимальный префиксный код C_{k-1} для Λ_{k-1}: $c_{k-1}(a_{k-1}) = c_k(\underbrace{a_{k-1}b_{k-1}})0, c_{k-1}(b_{k-1}) = c_k(\underbrace{a_{k-1}b_{k-1}})1, \forall x \in \Lambda_k, x \neq \underbrace{a_{k-1}b_{k-1}}\ c_{k-1}(x) = c_k(x)$
Так строим, пока не получим C_0 - оптимальный префиксный код для $\Lambda_0 = \Lambda$.
Пример:
$\Lambda = \{a, b, c, d, e, f, g\},\ p(a) = 0.13, p(b) = 0.08, p(c) = 0.25, p(d) = 0.18, p(e) = 0.03, p(f) = 0.12, p(g) = 0.21$
$a_0 = e, b_0 = b.$
$\Lambda_1 = \{a, \underbrace{eb}, c, d, f, g\},\ p(a) = 0.13, p(\underbrace{eb}) = 0.11, p(c) = 0.25, p(d) = 0.18, p(f) = 0.12, p(g) = 0.21$
$a_1 = \underbrace{eb}, b_1 = f\ \Lambda_2 = \{a, \underbrace{ebf}, c, d, g\},\ p(a) = 0.13, p(\underbrace{ebf}) = 0.23, p(c) = 0.25, p(d) = 0.18, p(g) = 0.21$
$a_2 = a, b_2 = d$
$\Lambda_3 = \{\underbrace{ad}, \underbrace{ebf}, c, g\},\ p(\underbrace{ad}) = 0.31, p(\underbrace{ebf}) = 0.23, p(c) = 0.25, p(g) = 0.21$
$a_3 = g, b_3 = \underbrace{ebf}$
$\Lambda_4 = \{\underbrace{ad}, \underbrace{gebf}, c\},\ p(\underbrace{ad}) = 0.31, p(\underbrace{gebf}) = 0.44, p(c) = 0.25$
$a_4 = c, b_4 = p(\underbrace{ad})$
$\Lambda_5 = \{\underbrace{cad}, \underbrace{gebf}\},\ p(\underbrace{cad}) = 0.56, p(\underbrace{gebf}) = 0.44$
Тогда $c_5(\underbrace{gebf}) = 0, c_5(\underbrace{cad}) = 1$. Теперь раскрываем алфавит обратно:
$c_4(\underbrace{gebf}) = 0, c_4(c) = 10, c_4(\underbrace{ad}) = 11$
$c_3(g) = 00, c_3(\underbrace{ebf}) = 01, c_3(c) = 10, c_3(\underbrace{ad}) = 11$
$c_2(g) = 00, c_2(\underbrace{ebf}) = 01, c_2(c) = 10, c_2(a) = 110, c_2(d) = 111$
$c_1(g) = 00, c_1(\underbrace{eb}) = 010, c_1(f) = 011, c_1(c) = 10, c_1(a) = 110, c_1(d) = 111$
$c_0(g) = 00, c_0(e) = 0100, c_0(b) = 0101, c_0(f) = 011, c_0(c) = 10, c_0(a) = 110, c_0(d) = 111$
$E\, l_0 = 2 * 0.21 + 4 * 0.03 + 4 * 0.08 + 3 * 0.12 + 2 * 0.25 + 3 * 0.13 + 3 * 0.18 = 2.65$
Заметим также, что параллельно с построением кода можно строить соответствующее ему двоичное дерево. Тогда код - это набор путей из корня в произвольную вершину в произвольном двоичном дереве, префиксный код - набор путей из корня в листья в произвольном двоичном дереве.

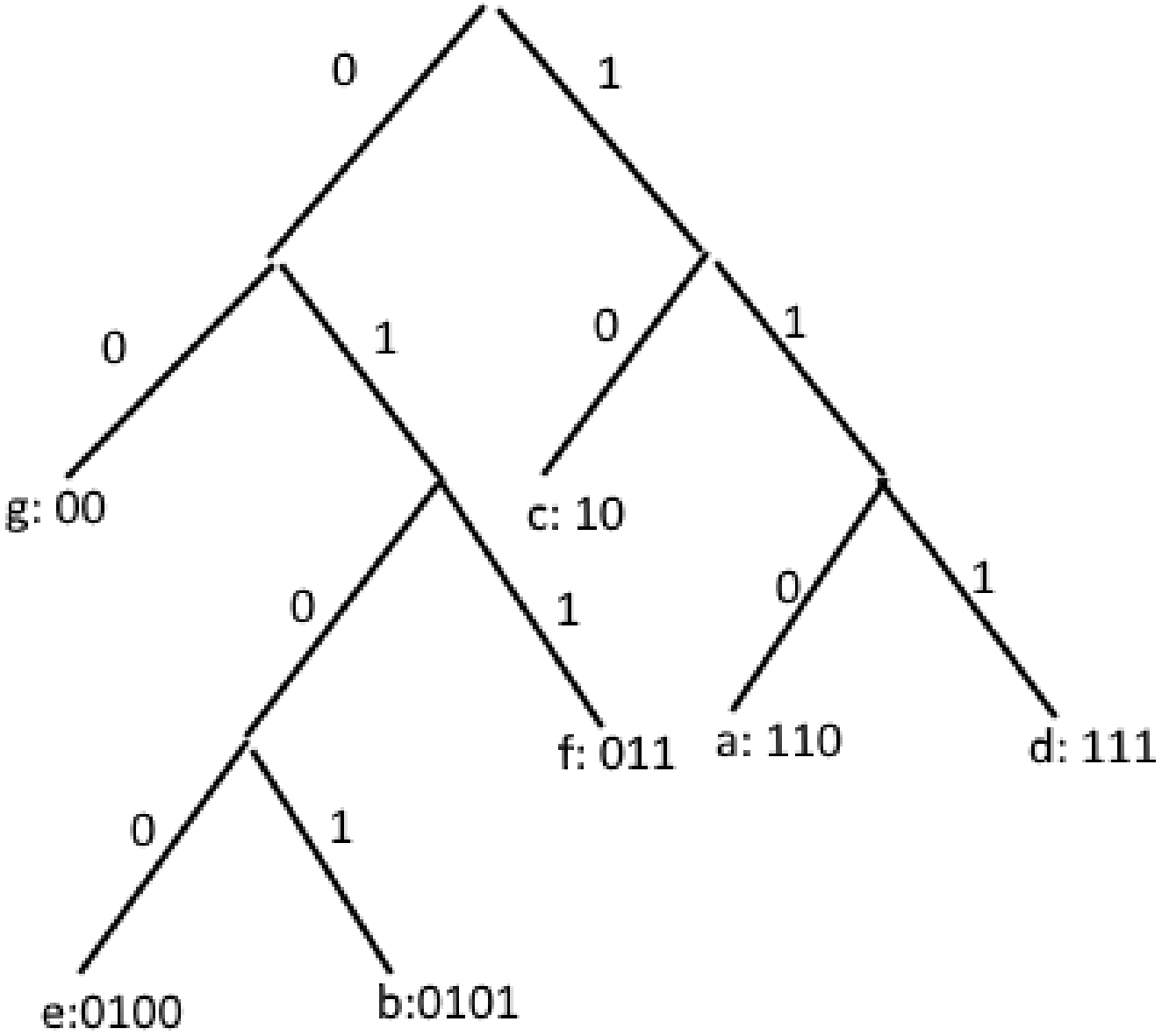

Рис. 10: Иллюстрация к примеру.

5.7 Самостоятельное изучение

5.7.1 Неравенство Крафта.

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

Теорема. Для того чтобы набор длин $l_1, \ldots, l_m$ мог быть набором длин кодовых последовательностей некоторого префиксного кода для алфавита из m символов необходимо и достаточно, чтобы $\sum\limits_{i \in 1:m} 2^{-l_i} \leqslant 1$.

Доказательство. Необходимость. Существует некоторый префиксный код для алфавита мощности m, кодовые последовательности которого имеют длины $l_1, \ldots, l_m$. Проверим выполнение неравенства Крафта. Обратимся к нашей интерпретации множества кодовых последовательностей как набор всех путей на двоичном дереве от корня к листу. Исходный, корневой, узел назовём нулевым уровнем, а дальше – последовательно прибавляем номер уровня по мере удаления от нулевого. Каждому узлу v на t-м уровне поставим в соответствие число $a(v) = 2^{-t}$ (тогда, например, корню соответствует $2^{-0} = 1$).

Пусть мы нашли узел v на уровне t, не являющийся листом, то есть на уровне $t+1$ есть хотя бы один узел, который получился после ветвения из данного – назевём их $N(v)$ (это множество, состоящее из одного или двух элементов). Тогда, $a(v) \geqslant \sum\limits_{u \in N(v)} a(u)$ — неравенство, если один узел, равенство, если два узла.

Просуммируем такие неравенства для каждого не листа:

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

.

$$2^0 \geqslant \sum_{u \text{ листья}} a(u).$$

Обратно, пусть выполнено неравенство. Пусть $l_1 \leqslant \ldots \leqslant l_m$.

n_j — число листьев, которые должны оказаться на уровне j: $n_j = |\{i : l_i = j, i \in 1:m\}|$.

Известно,

$$\sum_{i \in 1:m} 2^{-l_i} \leqslant 1,$$

значит,

$$\sum_{j \in 1:l_m} 2^{-j} n_j \leqslant 1.$$

Тогда, для каждого $j \in 1:l_m$,

$$n_j \leqslant 2^j - (2^{j-1} n_1 + \ldots + 2 n_{j-1}).$$

Пусть $m \neq 1$ (случай одной вершины очевиден), рассмотрим n_1, выделим на первом уровне вершин $n_1 \leqslant 2$, на втором уровне останется $2(2 - n_1)$. Известно, что $n_2 \leqslant 2^2 - 2n_1$, значит, осталось не меньше, чем требуется для второго уровня.

Пусть на $(j-1)$-м уровне было свободно

$$2^{j-1} - (2^{j-2} n_1 + \ldots + 2 n_{j-2}),$$

известно, что n_{j-1} не больше этой величины. Выделим n_{j-1} узлов, останется $2^{j-1} - (2^{j-2}n_1 + \ldots + 2n_{j-2}) - n_{j-1}$, значит, на j-м уровне будет $2 * (\ldots) = 2^j - (2^{j-1}n_1 + \ldots + 2n_{j-1})$.
Таким образом, строится двоичное дерево с нужным количеством листьев на нужных уровнях, по которому восстановим префиксный код.

5.8 15.02.19

Примечание: здесь и далее (в том числе, в последующих лекциях) при введении условной вероятности мы подразумеваем, что нужные вероятности положительны, поэтому это условие будет опускаться.

5.8.1 Конечная случайная схема и энтропия

Пусть $A_1, A_2, \ ... \ , A_n$ - разбиение множества исходов S вероятностного пространства (S, Pr). Конечной случайной схемой (КСС) называется схема α, сопоставляющая каждому A_i вероятность $Pr(A_i)$.

Энтропией КСС называется $H(\alpha) = -\sum\limits_{i=1}^{n} Pr(A_i) * \log Pr(A_i)$.

Некоторые свойства энтропии:

- $H(\alpha) \geq 0$

- Энтропия характеризует неопределенность, заключенную в КСС

- Для любой КСС α, у которой k исходов, $H(\alpha) \leq \log k$
  Доказательство:
  Обозначим $f(x) = -x * \log x$. На отрезке $[0, 1]$ $f(x)$ строго вогнутая, а значит, по неравенству Йенсена,
  $\sum\limits_{i=1}^{n} \lambda_i * f(x_i) \leq f(\sum\limits_{i=1}^{n} \lambda_i * x_i)$,
  Причем равенство достигается тогда и только тогда, когда $x_1 = \ ... \ = x_n$.
  Тогда если взять $x_i = Pr(A_i)$ и $\lambda_i = \frac{1}{k}$ $\forall i \in 1..k$, получаем:
  $\sum\limits_{i=1}^{k} \frac{1}{k}(-Pr(A_i) * \log Pr(A_i)) \leq -\sum\limits_{i=1}^{k} \frac{1}{k} Pr(A_i) * \log(\sum\limits_{i=1}^{k} \frac{1}{k} Pr(A_i))$
  $-\frac{1}{k} \sum\limits_{i=1}^{k} Pr(A_i) * \log Pr(A_i) \leq -\frac{1}{k} \log \frac{1}{k}$
  $-\sum\limits_{i=1}^{k} Pr(A_i) * \log Pr(A_i) \leq \log k$

- Из предыдущего пункта и условия равенства для неравенства Йенсена следует, что максимально возможную энтропию для КСС с k исходами, равную $\log k$, имеет схема с k равновероятными исходами.

- $H(\alpha) = 0 \Leftrightarrow \exists!$ достоверный исход в α.
  Доказательство в обе стороны очевидно. Если в сумме, составляющей энтропию, есть хоть одно ненулевое слагаемое, то и энтропия ненулевая, а $x \log x = 0$ только при $x = 0$ и $x = 1$.

5.8.2 Энтропия пересечения и условная энтропия

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

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

Воспользовавшись тем, что $Pr(A_i \cap B_j) = Pr(A_i) * Pr(B_j|A_i)$, получаем

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

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

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

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

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

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

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

Таким образом, в итоге формулу можно записать как

$H(\alpha \cap \beta) = H(\alpha) + H_\alpha(\beta)$.

Докажем, что $0 \le H_\alpha(\beta) \le H(\beta)$:

Левая часть очевидна из тех же соображений, из которых следует неотрицательность энтропии.

чтобы доказать правую, напишем для зафиксированного j, функции $f(x) = -x * \log x$, $\lambda_i = Pr(A_i)$, $x_i = Pr(B_j|A_i) \ \forall i \in 1..k$ неравенство Йенсена:

$\sum_{i=1}^{k} Pr(A_i) * (-Pr(B_j|A_i) * \log Pr(B_j|A_i)) \le -(\sum_{i=1}^{k} Pr(A_i) * Pr(B_j|A_i)) * \log \sum_{i=1}^{k} Pr(A_i) * Pr(B_j|A_i)$

Преобразуем правую часть:

$-(\sum_{i=1}^{k} Pr(A_i) * Pr(B_j|A_i)) * \log \sum_{i=1}^{k} Pr(A_i) * Pr(B_j|A_i) = -(\sum_{i=1}^{k} Pr(B_j \cap A_i)) * \log \sum_{i=1}^{k} Pr(B_j \cap A_i) =$

$-Pr(B_j) * \log Pr(B_j)$

После чего просуммируем обе части неравенства по j:

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

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

$\sum_{i=1}^{k} Pr(A_i) * H(\beta|A_i) \le H(\beta)$

$H_\alpha(\beta) \le H(\beta)$.

Из условия равенства для неравенства Йенсена следует, что $H_\alpha(\beta) = H(\beta) \Leftrightarrow$ все $Pr(B_j|A_i)$ равны между собой.

По формуле полной вероятности,
$\forall j \in 1..l\ Pr(B_j) = \sum_{i=1}^{k} Pr(B_j|A_i) * Pr(A_i)$
Используя равенство всех $Pr(B_j|A_i)$ пишем:
$\forall j \in 1..l\ Pr(B_j) = Pr(B_j|A_1) * \sum_{i=1}^{k} Pr(A_i)$
$\forall j \in 1..l\ Pr(B_j) = Pr(B_j|A_1)$
То есть, $\forall i \in 1..k, \forall j \in 1..l\ Pr(B_j) = Pr(B_j|A_i)$ Вспомним теперь определение взаимно независимых событий:
A и B независимы $\Leftrightarrow Pr(A \cap B) = Pr(A) * Pr(B) \Leftrightarrow Pr(A) * Pr(B|A) = Pr(A) * Pr(B) \Leftrightarrow Pr(B|A) = Pr(B)$.
КСС α и β называются независимыми, когда все исходы α независимы со всеми исходами β.
В таком случае, $H_\alpha(\beta)$ максимальна и равна $H(\beta)$.

5.8.3 Количество информации

Величина $I(\alpha, \beta) = H(\beta) - H_\alpha(\beta)$ называется количеством информации.
Запишем несколько свойств количества информации, которые доказываются простой проверкой:

- $I(\alpha, \beta) \geq 0$
- $I(\alpha, \beta) = H(\beta) \Leftrightarrow H_\alpha(\beta) = 0$
- $I(\alpha, \beta) = I(\beta, \alpha)$
- $I(\alpha, \beta) = 0 \Leftrightarrow \alpha$ и β независимы.

Пример:
Загадано натуральное число $x \in 1..N$
β - опыт, состоящий в нахождении x, β_m - опыт, показывающий, делится ли x на m, $m \in 1..N$.
У β N исходов, у β_m два исхода.
$H_{\beta_m}(\beta) = Pr\{x \vdots m\} * H(\beta|"x \vdots m") + Pr\{x \not\vdots m\} * H(\beta|"x \not\vdots m")$
Если обозначить количество чисел от 1 до N, которые делятся на m, как $q = \lfloor \frac{N}{m} \rfloor$, то мы получим:
$Pr\{x \vdots m\} = \frac{q}{N}$
$Pr\{x \not\vdots m\} = \frac{N-q}{N}$
$H(\beta|"x \vdots m") = -\sum_{i \vdots m, i \in 1..N} \frac{1}{q} * \log \frac{1}{q} = -\frac{q}{q} * \log \frac{1}{q} = \log q$
Аналогично $H(\beta|"x \not\vdots m") = \log(N - q)$
Таким образом, $H_{\beta_m}(\beta) = \frac{q}{N} * \log q + \frac{N-q}{N} * \log(N - q)$
$I(\beta_m, \beta) = \log N - \frac{q}{N} * \log q - \frac{N-q}{N} * \log(N - q) =$

$= \frac{q}{N} * \log N - \frac{q}{N} * \log q + \frac{N-q}{N} * \log N - \frac{N-q}{N} * \log(N-q) =$
$= -\frac{q}{N} * \log \frac{q}{N} - \frac{N-q}{N} * \log \frac{N-q}{N} \leq \log 2$
Равенство достигается при $q = N - q = \frac{N}{2}$, то есть если N четно и $m = 2$.

5.9 22.02.19

5.9.1 Пример с данетками

"Say 'what' again, I dare you, I dare you, I double-dare you, say 'what' one more goddamn time!"Итак, загадано число от 1 до N, опыт β - угадать число, опыт α - задать любой общий (да/нет) вопрос и получить ответ.
$H(\beta) = \log N$, поскольку с равной вероятностью было загадано каждое из N чисел.
$H(\alpha) \leq \log 2$, поскольку есть всего 2 варианта ответа.
Тогда $H(\alpha_1\alpha_2...\alpha_k) \leq \log 2^k = k \log 2$ (k вопросов, на каждый 2 варианта ответа).
А значит, чтобы угадать число, потребуется $k \geq \frac{\log N}{\log 2} = \log_2 N$ вопросов.
Есть ли какой-то алгоритм, который умеет-таки угадывать загаданное число за $O(\log N)$? Да. Называется бинпоиск (он же бинарный поиск, двоичный поиск).

5.9.2 Пример с избыточным кодированием

Итак, есть сообщение $u \in \{0,1\}^k$, которое нужно передать. При этом мы можем передать сообщение $x(u) \in \{0,1\}^n, n \geq k$, содержащее некоторую избыточную информацию. Зачем? Затем что канал связи "шумит"и может допускать ошибки. Конкретно, не более d ошибок на сообщение.
Итак, есть два сту... опыта:
β заключается в нахождении всех d ошибок. Сколько у β исходов? Для каждого количества ошибок j от 0 до d есть $\binom{n}{j}$ вариантов их расположения, то есть всего исходов у β $\sum\limits_{j=0}^{n}\binom{n}{j}$, откуда
$H(\beta) = \log \sum\limits_{j=0}^{n}\binom{n}{j}$
α - это наше дополнительное сообщение размера $n-k$, соотвественно, таких сообщений всего 2^{n-k}, а значит, $H(\alpha) = \log 2^{n-k} = (n-k)\log 2$.
Таким образом, чтобы суметь гарантированно найти все ошибки, нужно, чтобы $H(\alpha) \geq H(\beta)$. Отсюда

$$(n-k)\log 2 \geq \log \sum_{j=0}^{n}\binom{n}{j}$$
$$n-k \geq \log_2 \sum_{j=0}^{n}\binom{n}{j}$$
$$n \geq k + \log_2 \sum_{j=0}^{n}\binom{n}{j}$$
$$k \leq n - \log_2 \sum_{j=0}^{n}\binom{n}{j}$$

Это значит, что если наш канал связи допускает не более d ошибок, чтобы передать сообщение размера k нам понадобится не менее $k + \log_2 \sum\limits_{j=0}^{n}\binom{n}{j}$ бит.
Или, что более естественно, поскольку количество ошибок обычно таки зависит от размера переданного сообщения (а еще потому что правая часть зависит от n, и выразить k значительно проще), если мы передаем n бит, и из них не более d могут быть ошибочными, в переданном сообщении можно закодировать сообщение длиной не более $n - \log_2 \sum\limits_{j=0}^{n}\binom{n}{j}$.

5.9.3 Код Хэминга

Частный случай предыдущей задачи при $d = 1$. Итак, как мы уже выяснили, $2^{n-k} \geq \sum_{j=0}^{1} \binom{n}{j} = 1 + n$. Сделав замену переменной $l = n - k$ (длина "избыточного"сообщения) получим $k \leq 2^l - l - 1$.

Таблица 1: Максимальные значения k для фиксированных l

l	k
1	0
2	1
3	4
4	11
5	26
6	57

Как видно из таблицы, на совсем маленьких сообщениях все плохо, но чем больше сообщение, тем меньше (относительно) нужно лишней информации. Но как же так хитро передавать дополнительную информацию?
Итак, пусть $k = 12$ и мы хотим передать сообщение $u = 101101011100$. Нарисуем таблицу, j-м столбцом которой будет двоичная запись числа j (от младшего бита к старшему), размера 17 (по нашей таблице, для 12 бит сообщения 4 бит дополнительной информации уже мало, и надо брать 5, а $12 + 5 = 17$).

Таблица 2: Двоичная матрица A

l	0	1	0	l	0	1	0	l	0	1	0	l	0	1	0	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

А теперь зарезервируем в нашем сообщении длины 17 места с номерами 2^i (1, 2, 4, 8, 16), а на остальные позиции запишем наше сообщение:
$x_0(u) =$ __1_011_0101110_0
А теперь начинается магия: подберем на позицию 2^i такую цифру, чтобы произведение $x(u)$ и i-й строки матрицы было равно 0. Как же мы подберем, если в $x(u)$ еще куча неопределенных позиций? Да очень просто: на этих позициях в строчке с номером i стоят нули, потому что в двоичной записи числа 2^j единица стоит только на позиции j, а значит, при $i \neq j$ там стоит ноль, следовательно, что там будет стоять в итоговом виде $x(u)$ нам все равно. Отсюда же очевидно, что на позиции 2^i в i-й строчке будет всегда стоять единица.
Итак, подберем нужную цифру:
?∗1+_∗0+1∗1+_∗0+0∗1+1∗0+1∗1+_∗0+0∗1+1∗0+0∗1+1∗0+1∗1+1∗0+0∗1+_∗0+0∗1 =

$? + 1 + 1 + 1 = 1 + ? = 0$, откуда $? = 1$.
$x_1(u) = 1\_1\_011\_0101110\_0$
Аналогично заполняем остальные пропуски, получая $x(u) = 11110110010111000$.
Как же теперь понять, где была ошибка? Давайте испортим 10-ю позицию:
$y = 11110110000111000$
А теперь посчитаем $A \times y^T = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
Получили двоичную запись (старший бит снизу) позиции, в которой произошла ошибка: $2^1 + 2^3 = 10$. Как?
При умножении на i-ю строку матрицы j-я позиция сообщения влияла только если $A[i][j] = 1$, то есть если на i-м месте в двоичной записи числа j стояла единица. Поэтому результат произведения строки матрицы на столбец сообщения изменился (став единицей) только для тех строк, где на 10-й позиции стояла единица (а это строки с номерами, равными позициям, где в двоичной записи числа 10 стоят единицы), а для остальных строк остался нулем.