Домашнее задание 13
Шумилкин Андрей, группа 163

Задача 1

Пусть множество исходов – это последовательность, в которой родились дети и тогда все исходы равновероятны, так как рождение мальчика и девочки равновероятны. То, что мальчик родился в понедельник никак не влияет на вероятность, поскольку в тот же понедельник могла за ним родиться девочка или наоборот(двойняшки) и по условию рождение каждого – равновероятно. Обозначим М – мальчик, Д – девочка. Тогда множество исходов имеет вид $\{\text{ММ}, \text{МД}, \text{ДМ}, \text{ДД}\}$ и вероятность каждого исхода равна $1/4$. С учетом того, что по условию один из детей должен быть мальчиком, а другой – девочкой, нам пододят два из этих исходов – МД и ДМ, и тогда итоговая вероятность равна $1/2$.
Ответ: $\frac{1}{2}$.

Задача 3

Пусть множество исходов – это пятерки выбранных чисел и все исходы равновероятны.
Количество всех исходов равно C_{36}^5. Вероятность выбрать пятерку, среди элементов которой есть число 2 будет равна $\frac{C_{35}^4}{C_{36}^5}$, так как пятерка должна быть выбрана по условию и еще четыре элемента мы выбираем из оставшихся 35-и элементов множества. Вероятность выбрать пятерку, среди элементов которой будет число 5 так же равна $\frac{C_{35}^4}{C_{36}^5} = \frac{5}{36}$.
Теперь посчитаем вероятность выбрать элемент 5 при том, что двойка уже выбрана. Вероятность выбрать пятерку в которой есть элементы и 5, и 2 равна $\frac{C_{34}^3}{C_{36}^5}$.
Тогда вероятность выбрать элемент 5 при том, что элемент 2 уже выбран равна $\frac{C_{34}^3}{C_{36}^5} \cdot \frac{C_{36}^5}{C_{35}^4} = \frac{C_{34}^3}{C_{35}^4} = \frac{4}{35}$.
И, так как $\frac{5}{36} \neq \frac{4}{35}$, то есть вероятность события не равна вероятности его же при условии какого-то другого делаем вывод, что события зависимы.
Ответ: Данные события зависимы.

Задача 4

Пусть множество исходов – это какая-то определенная функция и все исходы равновероятны. Всего вариантов составить какую-либо функцию у нас будет n^n, поскольку она всюду определена и каждому из n элементов одного множества мы можем и должны сопоставить один из n элементов другого множества.
Составить же инъективную функцию у нас $n!$ способов, то есть мы будем элементам первого множества сопоставлять некую перестановку из элементов второго. Значит вероятность того, что функция инъективна равна $\frac{n!}{n^n}$.
Составить функцию так, чтобы $f(1) = 1$ у нас будет $(n-1)^n$ способов, то есть один

элемент мы определяем изначально, а для остальных $n-1$ выбираем один из n элементов другого множества. Тогда вероятность того, что $f(1)=1$ будет $\frac{1}{n}$
Если же хотим получить инъективную функцию у которой $f(1)=1$, то у нас будет $(n-1)!$ способов сделать это, поскольку мы определяем $f(1)=1$ и нам нужно для оставшихся $n-1$ элемента сопоставить различные элементы другого множества, в которое уже не входит 1, так как мы уже составили с ней пару.
Тогда вероятность того, что $f(1)=1$, при условии, что функция инъективна равна $\frac{(n-1)!}{n^n} \cdot \frac{n!}{n^n} = \frac{1}{n}$.
Можно заметить, что она равна вероятности того, что $f(1)=1$, а значит данные события независимы.
Ответ: Данные события независимы.

Задача 5

Пусть множество исходов –
Обозначим правильное решение как 1, а неправильное – как 0. Рассмотрим все возможные исходы выбора первыми двумя членами жюри. Это: $\{00, 01, 10, 11\}$.
Теперь посчитаем вероятность того, что третий член жюри сделает правильный выбор для каждого случая:

1. 00. Итоговая вероятность равна 0, так как правильного решения тут нет.
2. 01. Вероятность такого случая равна $(1-p)\cdot p$, тогда вероятость правильного выбора третьим членом жюри равна $\frac{p\cdot(1-p)}{2}$.
3. 10. Вероятность такого случая равна $p\cdot(1-p)$, тогда вероятость правильного выбора третьим членом жюри равна $\frac{p\cdot(1-p)}{2}$.
4. 11. Вероятность такого случая равна p^2, тогда вероятость правильного выбора третьим членом жюри равна p^2.

А общая вероятность выбора правильного решения третьим из судей будет равна сумме данных вариантов:
$p\cdot(1-p)+p^2 = p^2+p-p^2 = p$.
Видно, что эта вероятность равна p – вероятности правильного решения, принимаемого одним добросовестным членом жюри. **Ответ:** Вероятность выбора верного решения равна p и равна вероятности правильного решения, принимаемого одним добросовестным членом жюри.

Задача 7

Посчитаем вероятность как бы спускаясь вниз, при этом вероятность выигрыша при счете 10:9 возьмем, конечно же, за 1.
Тогда вероятность выигрыша при 9:9 $= 1/2$.
9:8 – это будет сумма вероятностей вариантов (выиграть сразу) и (проиграть, а потом выиграть) и она равна $1/2\cdot 1/2 + 1/2 = 3/4$.

9:7 – это будет сумма вероятностей вариантов (выиграть сразу) и (перейти в случай 9:8), для которого мы уже посчитали вероятность и тогда она будет равна $1/2 + 1/2 \cdot 3/4 = 7/8/$. 8:9 – для выигрыша можем перейти в вариант 9:9 для которго уже посчитали, тогда равна $1/2 \cdot 1/2 = 1/4$. 8:8 – перйти в вариант 9:8, либо 8:9, тогда равна $1/2 \cdot 3/4 + 1/2 \cdot 1/4 = 1/2$. 8:7 – перейти в вариант 9:7, либо в вариант 8:8. Тогда вероятность равна $1/2 \cdot 1/2 + 1/2 \cdot 7/8 = 11/16$.
Ответ: $11/16$.

Вероятность A при условии B:

$$Pr[A|B] = \frac{Pr[A \cap B]}{Pr[B]}.$$

$$Pr[A|B] = Pr[A] \cdot \frac{Pr[B|A]}{Pr[B]}.$$

Домашнее задание 14
Шумилкин Андрей, группа 163

Задача 1

Раз на выирыши уходит 40%, то есть это «то, что мы будем получать в среднем, если будем повторять эксперимент много раз» $=>$ $E[x] = 40$. Тогда по неравенству Маркова $Pr[x \geq 5000] \leq \frac{40}{5000} \leq 0,008$, то есть вероятность выиграть 5000 и больше, меньше или равна 0,008 что меньше 0,01.

Задача 2

В среднем люди жили 26 лет, то есть можем говорить, что математическое ожидание равно 26. Рассмотрим два крайних случая, когда люди, которые жили мало: когда они проживали 0 лет и когда проживали восемь. По условию прожить меньше 9 лет равна $1/2$. Тогда прожить больше так же будет $1/2$.
Отсюда для 0 получаем: $1/2 \cdot 0 + 1/2 \cdot t_1 = 26$, где t_1 – средний возраст проживших более 8 лет, когда жившие мало жили 0 лет в среднем. Тогда $t_1 = 52$.
И для 8 получаем: $1/2 \cdot 8 + 1/2 \cdot t_2 = 26$, где t_2 – средний возраст проживших более 8 лет, когда жившие мало жили 8 лет в среднем. Тогда $t_2 = 44$.
И раз мы рассмотрели крайние значения, то мы получили границы искомого интервала: 44 и 52.

Задача 3

Как известно, мат. ожидание честной кости равно 3,5, тогда у первого игрока мат. ожидание, то есть и средний выигрыш будет равен 12,25. А у второго средний выигрыш равен $\frac{1^2+2^2+3^2+4^2+5^2+6^2}{6} = \frac{91}{6}$, что примерно равно 15,1 и больше, чем 12,25. Значит средний выигрыш второго игрока больше, чем у первого.

Задача 4

Пусть у нас будет некоторая индикаторная величина, которая обозначает начинается ли с данного элемента подстрока ab. Заметим, что ее мат. ожидание равно $1/4$, потому что возможных строк всего 4 – $\{1, 2, 3, 4\}$.
Тогда мы можем представить нашу функцию как сумму индикаторных величин, при этом заметим, что позиций с которых может начинаться строка длины 2 всего 19.
И мы знаем, что математическое ожидание мы тогда можем представить как сумму математических ожиданий индикаторных величин, которых всего 19 и значит искомое мат. ожидание равно $19/4$.

Задача 5

Пусть у нас есть некоторая индикаторная велиина, которая обозначает, что завтрак попробован. Вероятность того, что завтрак попробован будет $1 - \frac{9^{15}}{10^{15}}$ и ее математическое ожидание тому же.
Тогда мы можем представить среднее кол-во попробованных завтраков как сумму математических ожиданий индикаторных величин. Их всего 10 и тогда искомое математическое ожидание равно $\frac{9^{15}}{10^{14}}$.

Домашнее задание 15
Шумилкин Андрей, группа 163

Задача 1

Как мы знаем, если некоторое множество U бесконечно, а множество V конечно или счетно, то $U \cup V$ равномощно U.
Обозначим $C = A \setminus B$. Тогда мы можем записать $A = C \cup B$. Тогда по свойству, упомянотому выше мощность C равна мощности A и C не может быть не бесконечным, поскольку объединение счетного множества со счетным будет не более, чем счетным.
Значит все утверждение верно.

Задача 2

Заметим, что $A \triangle B = (A \setminus B) \cup (B \setminus A)$. Как мы доказали в прошлой задаче $A \setminus B$ равномощно A. А $(B \setminus A)$ будет не более чем счетно, так как само B счетно.
И тогда из свойства, упомянутого в первой задаче и из транзитивности равномощности заметим, что раз $A \setminus B$ равномощно A и объединение бесконечного множества U и счетного множества V будет равномощно U, что $A \triangle B$ равномощно A.
Значит все утверждение верно.

Задача 3

Как мы знаем, если некоторое множество U бесконечно, а множество V конечно или счетно, то $U \cup V$ равномощно U.
Обозначим $C = A \setminus B$. Тогда мы можем записать $A = C \cup B$. Тогда по свойству, упомянотому выше мощность C равна мощности A и C не может быть не бесконечным, поскольку объединение конечного множества со счетным будет не более, чем счетным.
Значит все утверждение верно.

Задача 4

Как нам известно множество рациональных чисел счетно и в каждом интервале найдется хотя бы одно рациональное число по аксиоме полноты.
Тогда мы можем сопоставить кажому интервалу минимальную рациональную точку, находящуюся в нем. Такое сопоставление будет взаимно-однозначным, поскольку интервалы не пересекаются.
Раз мы смогли найти такое сопоставление, то мощность множества данных интервалов не больше, чем мощность множества рациональных чисел, а оно счетно. Значит множество интервалов будет не более чем счетно.

Задача 5

Как мы знаем, всякое бесконечное множество содержит счетное подмножество.
Это подмножество равномощно $\mathbb{N}$. Мы можем заметить, что и само $\mathbb{N}$ содержит бесконечное счетное число счетных подмножеств. Такими подмножествами будут, например, являться степени простых чисел, поскольку, как известно, простых чисел бесконечное число, а их степень так же принадлежит $\mathbb{N}$. И при этом они не будут пересекаться по основной теореме арифметики.
Значит из первичной биекции некоторого подмножества нашего множества в $\mathbb{N}$ мы можем выделить сопоставления со множествами различных простых чисел в степени, которые являются счетными и которых бесконечное число, что и будет означать, что всякое бесконечное множество содержит бесконечное число непересекающихся счетных подмножеств.

Домашнее задание 16
Шумилкин Андрей, группа 163

Задача 1

Заметим, что множество вещественных положительных чисел и ноль континуально , поскольку его подмножеством является интервал [0,1], который имеет мощность континуум.
Мы можем каждый круг охарактеризовать тройкой чисел (x, y, r), то есть его координатами центра и радиусом, при этом видно, что для разных кругов эта характеристика будет разной.
Мы можем строить круг в любой точке плоскости и с любым радиусом, значит все три числа примут всевозможные значения из множества положительных вещественных чисел и нуля.
И, как нам известно, $\mathbb{R}^3$ равномощно $\mathbb{R}$, откуда и следует, что множество всех кругов на плоскости континуально.

Задача 2

Нет, неверно, поскольку мы можем выбрать какую-либо точку и построить континуум окружностей с центром в ней и которые имеют радиусы, к примеру, которые равны всем точкам из отрезка [0,1]. Множество таких окружностей будет континуально, поскольку множество всех точек отрезка [0,1] континуально, но множество их центров будет иметь мощность один, так как мы по построению сделали все центры в одной точке.

Задача 3

Да, существует.
Мы знаем, что $\mathbb{R}^2$ равномощно $\mathbb{R}$, а в $\mathbb{R}^2$ мы можем найти такое семейство – это множество параллельных оси x прямых, которые характеризуются $y = c$. Каждая прямая континуальна и их множество тоже континуально, так как c может быть любым из $\mathbb{R}$. И, так как $\mathbb{R}^2$ равномощно $\mathbb{R}$, существует инъекция из $\mathbb{R}^2$ в $\mathbb{R}$ мы каждой прямой можем сопоставить некоторое число и они не будут пересекаться, так как и сами прямые не пересекаются.

Задача 4

Оно будет иметь мощность точно не больше мощности континуума, потому что мы можем любую последовательность перевести из двоичной системы счисления в десятичную и получить некоторое число, которое точно принадлежит R, а R имеет мощность континуума.
Воспользуемся теоремой Кантора-Берштейна.

Инъекция из множества двоичных последовательностей без трех подряд идущих единиц в обычные двоичные последовательности понятна – мы можем просто переводить в те же числа.
Теперь построим инъекцию из множества обычных двоичных послдеовательностей в последовательности без трех единиц подряд. Заметим, что единиц подряд тогда может быть одна или две. Тогда будем "переводить"наше число следующим образом: если на текущей позиции "0 то пишем одну единицу и за ней ноль, а если "1 то две единицы и за ними ноль.
Тогда в итоге получим последовательность без трех единиц подряд, соответствующую обычной двоичной последовательности, при том для разных они будут разные.
По теореме Кантора-Берштейна получаем, что наше множество равномощно множеству обычных двоичных последовательностей, а оно континуально, откуда следует, что наше множество так же имеет мощность континуум.

Домашнее задание 17
Шумилкин Андрей, группа 163

Задача 1

Воспользуемся теоремой Кантора-Берштейна.
Составим инъективные отображения из нашей последовательности в последовательность двоичных слов и обратно.
В последовательности двоичных слов мы можем просто между каждыми двумя символами вставить цифру два и тогда мы получим последовательность, состоящию из 0, 1 и 2 в которой по построению никакие два символа не идут подряд, а так же различным двоичным последовательностям соответствуют различные последовательности из 0, 1 и 2.
Нашу же последовательность мы можем перекодировать следующим образом: 0 будем сопоставлять 10, 1 – 100 и 2 – 1000. Тогда на выходе мы получим обычную двоичную последовательность, при этом она однозначно будет раскодироваться в последовательность из 0, 1 и 2, что и значит, что мы построили инъекцию.
Тогда по теореме Кантора-Берштейна выходит, что мощность множества наших последовательностей равна мощности множества двоичных последовательностей $\Rightarrow$ оно континуально.

Задача 2

Заметим, что отношений эквивалентности не меньше континуума, поскольку если взять даже те отношения, которые разбивают множества на два клааса эквивалентности мы можем закодировать их некоторыми бесконечными двоичными последовательностями таким образом: возьмем первый элемент и все элементы эквивалентные ему, включая его самого обозначим 0, а остальные – 1.

Далее будем идти по элементам и записывать цифру, которую мы сопоставили этому элементу и таким образом будет получаться некая двоичная последовательность, множество которых, как нам известно, континуально.
Теперь заметим, что не больше, поскольку мы каждому отношению можем поставить в соответствие последовательность натуральных чисел, а множество таких последовательностей так же континуально.
Строить такое соответвствие будем следующим образом: первому элементу сопоставляем класс 1 и все эквивалентные ему элементы так же обозначаем 1. Далее находим первый необозначенный элемент и ставим ему в соответствие 2, а так же всем эквивалентным ему элементам. И т.д. n-ому необозначенному элементу ставим число n. В итоге получаем последовательность натуральных чисел.
Таким образом выходит, что наше множество континуально.

Задача 4

Соответствующая ДНФ будет иметь следующий вид:

$$(\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7 \wedge \neg x_8 \wedge x_9) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5 \wedge x_6 \wedge \neg x_7 \wedge x_8 \wedge \neg x_9) \vee$$

$$(\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7 \wedge \neg x_8 \wedge x_9) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5 \wedge x_6 \wedge \neg x_7 \wedge x_8 \wedge x_9) \vee$$

$$(\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge \neg x_6 \wedge x_7 \wedge x_8 \wedge x_9) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9) \vee$$

$$(\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9) \vee$$

$$(\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9) \vee (\neg x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9)$$

Задача 5

Мы можем выразить:

$$\neg X = X|X$$

$$X \vee Y = (X|X)|(Y|Y)$$

$$X \wedge Y = (X|Y)|(X|Y)$$

А мы знаем, что любое выражение можно выразить в виде ДНФ в котором как раз и применяются только эти три операции $\Rightarrow$ мы можем выразить любое выражение через штрих Шеффера $\Rightarrow$ система связок, состоящая только из штриха Шеффера обладает полнотой.

Задача 6

В качестве доказательства можно привести алгоритм приведения к КНФ, практически аналогичный алгоритму приведения к ДНФ, только эквивалентность представим по другому.
1. На первом шаге избавляемся от импликаций и эквивалентностей в выражении, представляя их в виде:

$$A \to B = \neg A \vee \neg B$$

$$A \leftrightarrow B = (\neg A \vee B) \wedge (A \vee \neg B)$$

2. Далее все знаки отрицания, относящиеся к выражениям заменим так, чтобы они относились к конкретным переменным:

$$\neg(A \vee B) = \neg A \wedge \neg B$$

$$\neg(A \wedge B) = \neg A \vee \neg B$$

3. Далее избавимся от всех знаков двойного отрицания.
4. Применяем там, где нужно, свойство дистрибутивности конъюкции и дизъюнкции.

В итоге получаем конъюкцию дизъюнкций переменных или их отрицаний.

Домашнее задание 18
Шумилкин Андрей, группа 163

Задача 1

Заметим, что данное выражение истинно, когда в нем либо одна переменная равна 1, либо три переменных равны 1.
Мы можем с помощью выражения $(x \wedge y \wedge z)$ записать ту часть выражения, которая истинна, когда все три переменных рывны 1.
Тогда нам остается записать часть выражения, которая истинна тогда и только тогда, когда лишь одна переменная равна 1.
Мы можем это сделать с помощью такого выражения: $(\neg((x \wedge y) \vee (x \wedge z) \vee (y \wedge z))) \wedge (x \vee y \vee z)$. То есть мы сначала проверяем, что в каэждой паре переменных хотя бы один равен нулю, откуда и получаем, что всего будет не больше одной переменной, равной единице. А потом проверяем, что она именно одна, а не ноль.
В итоге получаем:

$$(x \wedge y \wedge z) \vee (\neg((x \wedge y) \vee (x \wedge z) \vee (y \wedge z))) \wedge (x \vee y \vee z).$$

Задача 2

Выпишем 9 наборов переменных, на которых функция истинна:
{0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111}
Видим, что в восьми из них первая цифра истинна, а в одном – три последних. Заметим, что в оставшихся наборах, на которых функция не истинна такого нет, поэтому мы можем записать ее просто в виде $x_1 \vee (x_2 \wedge x_3 \wedge x_4)$.
Тогда схему можем записать как $g_1 = x_1,\ g_2 = x_2,\ g_3 = x_3,\ g_4 = x_4,\ g_5 = g_2 \wedge g_3,\ g_6 = g_4 \wedge g_5,\ g_7 = g_1 \wedge g_7$. Выход – g_7.

Задача 3

Мы можем строить схему по выражению вида: $(x_1 \wedge \neg x_2 \wedge x_3) \vee (x_2 \wedge \neg x_3 \wedge x_4) \vee \ldots \vee (x_{n-2} \wedge \neg x_{n-1} \wedge x_n)$.
Тогда мы можем добавить на первом "уровне"2n вершин в наш граф-схему: сами значения переменных $x_1, x_2, \ldots x_n$ и отрицания значений переменных $x_2, x_3, \ldots x_{n-1}$.
Тогда на втором уровне будем делать конъюкицию первых двух переменных в каждой из скобок, то есть $(x_1 \wedge \neg x_2), (x_2 \wedge \neg x_3), \ldots (x_{n-2} \wedge \neg x_{n-1})$ – на нем у нас будет n вершин.
Тогда на третьем уровне будем делать конъюкцию значений в вершинах, полученных на втором уровне с третьей переменной в скобках. На это так же потребуется n вершин.
После чего у нас останется n значений и нам нужно будет проверить равно ли хотя бы одно из них 1. Тогда мы просто сделаем дизъюнкцию первого значения со вторым, далее полученного значения с третьим и так далее. На выходе у нас как раз будет 1, если в исходном слове было подслово 101 и на эти дизъюнкции у нас уйдет n вершин.
Тогда размер полученной схемы будет (из рассчета 2n + n + n + n) – $O(6n) = O(n)$.

Задача 4

Будем представлять двоичное число из n разрядов как набор переменных $x_{n-1}, x_{n-2}, \ldots x_0$.
Сначала поймем, что простой путь умножить число на три в двоичном представлении – это сначала удвоить его, а потом прибавить к результату его самого.
Удвоение числа делается достаточно просто – оно сдвигается влево на 1 бит.
Теперь разберемся со сложением. Заметим, что при сложении двух данных чисел у нас получиться число с $n + 1$ разрядом. Обозначим их как $y_n, x_{n-1}, \ldots y_0$.
Начнем складывать. $y_0 = x_0$, поскольку второе число сдвинуто влево в этом разряде у него 0.
Чтобы продолжить складывать нам нужно ввести дополнительную переменную, назовем их z, в которой мы будем хранить то число, которое будет переходить в следующий разряд. Т.е. $y_1 = x_1 \vee x_0$ и $z_1 = x_0 \wedge x_1$ и тогда $y_2 = x_2 \vee x_1 \vee z_1$.
Тогда далее имеем: $y_i = x_i \vee x_{i-1} \vee z_{i-1}$,
$z_i = (x_{i-1} \wedge x_i) \vee (x_{i-1} \wedge z_{i-1}) \vee (x_i \wedge z_{i-1})$.
И такое упрощение, а точнее то, что мы не повторяем вычисление пременных, а уже используем вычисленные раннее позволяет нам построить схему полиномиального размера.

Домашнее задание 19
Шумилкин Андрей, группа 163

Задача 1

Если граф задан матрицей смежности, то нам достаточно просто для каждой вершины подсчитать дизъюнкцию всех переменных в ее строке, кроме той, что стоит на диагонали и тогда мы получим для каждой вершины переменную равную единице только тогда, когда данная вершина соединенена с какой-то другой.
Тогда вторым шагом нам достаточно подсчитать конъюкцию всех этих вершин, которая будет равна нулю, когда хотя бы одна вершина не соединена ни с какими другими вершинами. Значит нам нужно просто взять отрицание подсчитанной конъюкции.
И так как мы просто один раз просматриваем матрицу смежности, размер которой n^2, то схема получится полиномиальной.

Задача 2

Заметим, что выбрать в графе три вершины мы можем C_n^3 способами, т.е. $\frac{n(n-1)(n-2)}{6}$ и данное выражение является полиномом.
Тогда будем выбирать такие тройки, делать конъюнкцию элементов, стоящих в них, которая будет равна единице тогда, когда они образуют треугольник. Тогда отрицание дизъюнкции конъюнкций всех троек как раз и вернет ответ, поскольку просто дизъюнкция конъюкций всех троек будет равна единице как раз тогда, когда в графе есть один треугольник.
И так как троек всего полиномиальное количество, то и схема получится полиномаилального размера.

Задача 3

Чтобы в графе существовал Эйлеров цикл он должен быть связен и все вершины в нем должны быть четной степени.
Четность степени всех вершин мы можем проверить для каждой вершины подсчитав xor(который так же называют сложением по модулю 2) всех переменных в ее строке матрицы смежности, кроме той, что стоит на диагонали. Заметим, что это значение будет равно нулю, если кол-во единиц будет четным. Тогда нам достаточно взять конъюкцию отрицаний данных значений для всех вершин и значение данного выражения будет равно единице, когда степени всех вершин в графе четны. Размерность этой схемы так же будет полиномиальна, так как мы просто раз просматриваем матрицу смежности.
А то, что можно проверить связность графа схемой, глубиной не больше $O(log^2 n)$ и то, как это сделать с помощью булевых степеней матрицы смежности мы рассматривали на лекции, тогда достаточно просто сделать конъюкцию получившихся из двух данных схем значений и, так как они обе имеют полиномиальный размер, то и итоговая схема так же будет полиномиальна.

Задача 4

Мы знаем, что любую функцию можно записать с помощью конъюкций и дизъюнкций просто представив ее в ДНФ.
Так же заметим, что всего функций будет 2^n. Тогда остается заметить, что СДНФ минимальной функции будет представима как дизъюнкция конъюкций, без отрицаний и размер такой схемы будет n.
Тогда размер общей схемы будет как раз $O(n * 2^n)$.

Домашнее задание 19
Шумилкин Андрей, группа 163

Задача 1

Если любая цифра числа π вычислима, то и любая пятерка подряд идущих цифр в числе π вычислима, т.е. мы моежм вычислять по числу из числа π и для всем встречающимся пятеркам сопоставлять единицу в некоторой функции, а тогда и их множество разрешимо по определению, т.к. множество называется разрешимым, если его характеристическая функция вычислима.
Хоть число π и бесконечно, а значит мы точно не знаем все пятерки цифр, которые могут в нем встречаться, но всего таких пятерок может быть лищь 10^5, а значит кол-во элементов в подмножестве, которое должен перечислить алгоритм также будет конечно. Любое конечное множество разрешимо и если множество разрешимо, то оно и перечислимо.

Задача 2

Раз мы можем перечислить элементы всего множества X, то мы точно так же можем перечислить элементы и необходимого подмножества X – чисел, сумма цифр которых равна 10.
Т.е. мы можем «идти» по X как при его перечислении, но включать в то множество, которое мы перечисляем только его элементы, соответсвующие условию.

Задача 3

Мы можем описать алгоритм перечисления такого множества: берем декартово произведние множеств $A \times B$ и каждый его элемент выводим. Тогда получается, что это вычилсимая функциция из множества $A \times B$ в некоторое множество его значений, а множество значений вычислимой функции – перечислимо.

Задача 4

По сути мы можем описать такой алгоритм, проверяющий некоторое свойство натуральных чисел: он на вход получает число и выводит и дает ответ 1(да), если данное число не принадлежит конечному подмножеству элементы которого не являются значениями рассматриваемой функции.
Тогда полученное множество будет по определению разрешимым, так как его характиристическая функция вычилсима. А разрешимое множество также является перечислимым.
А для непустого множество это равносильно тому, что оно является множеством значений некойвсюду определенной вычислимой функции, а значит рассматриваемая функция вычислима.

Домашнее задание 21
Шумилкин Андрей, группа 163

Задача 1

Поскольку существует функция, которая на всей области своео определения равна константе n и притом она вычислима, поскольку мы просто для любых входных данных выводим n, то для ее вычисления в нумерации универсальных функций есть некоторая программа, имеющая номер p, притом на элементе с тем же номером p ее значение будет равно равно n. Тогда и искомое множество будет совпадать с $\mathbb{N}$

Домашнее задание 22
Шумилкин Андрей, группа 163

Задача 1

Мы можем написать бесконечное количество различных программ, результатом работы которых при входных даных, равных некоторому x, будет 2017, потому что мы можем делать в самой программе хоть что(к примеру, комментарии, которые содержат произвольный текст приводят к тому, что возникает бесконечное множество таких программ), поэтому все они будут различны, но выводить константное число. Тогда, если мы занумеруем эти программы, как раз получится, что найдется бесконечно много подходящих p.

Задача 2

Введем функцию от двух аргументов – $V(n, x) = nx$. Видно, что она вычислима – достаточно просто домножить входные данные на константу n и также она всюду определена.
По свойству главных нумераций есть $V(n, x) = U(q(n), x) = nx$, то есть так же есть и необходимая всюду определенная вычислимая функция $q(n)$.

И тогда по теореме о неподвижной точке существует $U(q(n), x) = U(n, x) = nx$, поскольку функция $q(n)$ вычислима, что мы заметили из свойства главных нумераций.

Задача 3

По свойству главных нумераций есть $V(n, x) = U(q(n), x)$, то есть так же есть и необходимая всюду определенная вычислимая функция $q(n)$.

И тогда по теореме о неподвижной точке существует $U(q(p), x) = U(p, x)$, поскольку функция $q(n)$ вычислима, что мы заметили из свойства главных нумераций. Отсюда и выходит, что найдется такое p, что $V(p, x) = U(p, x)$

Домашнее задание 23
Шумилкин Андрей, группа 163

Задача 1

Как нам известно из тезиса Чёрна-Тьюринга всякая вычислимая функция вычислима машиной Тьюринга, а нигде не определенная функция является вычислимой, поскольку достаточно просто "зациклить"программу, то есть чтобы она уклонилась от выдачи какого-либо ответа – это и будет алгоритм ее вычисления.
Значит существует МТ, которая вычисляет нигде не определенную функцию.
Пусть A – входной алфавит и q – некотрый символ из $A \cup \{\Lambda\}$ Тогда нам достаточно просто зациклить МТ и мы можем сделать это следующей таблице переходов:

$$\delta :\mapsto \Big\{(q, 0) \mapsto (q(\text{тот же самый}), 0, +1)$$

Где 0 – начальное состояние.
Действительно, согласно таблице, машина сначала просто "пройдет"по входным данным в виде числа, а потом продолжит двигаться вправо по пустым символам бесконечно, так как лента машины бесконечна.

Задача 2

Приведем таблицу переходов:

$$\delta :\mapsto \begin{cases} (0,0) \mapsto (1,0,+1) \\ (1,0) \mapsto (0,0,+1) \\ (\Lambda,0) \mapsto (\Lambda,0,0) \end{cases}$$

Где 0 – начальное состояние.

Заметим, что если головка находится над символом из алфавита w, то есть над 0 или 1, то она сдвинется вправо, предварительно записав в предыдущую ячейку "отрицание"текущего символа, то есть инвертирует его.
Когда же головка дойдет до пробельного символа она просто остановится, при этом к этому моменту все символы входа из w будут инвертированы, что по определению, данному в задаче, и будет представлять инвертированное входное слово.

Задача 3

Пусть q – любая буква входного алфавита, а l – любой символ из символов входного алфавита и Λ.

Приведем таблицу переходов:

$$\delta :\mapsto \begin{cases} (a,0) \mapsto (a,1,+1) \\ (b,1) \mapsto (b,2,+1) \\ (a,2) \mapsto (a,3,+1) \\ (q,3) \mapsto (q,3,+1) \\ (\Lambda,3) \mapsto (\Lambda,4,-1) \\ (q,4) \mapsto (\Lambda,4,-1) \\ (\Lambda,4) \mapsto (1,5,+1) \\ \\ (a,1) \mapsto (a,0,+1) \\ (b,d \in \{0,2\}) \mapsto (b,0,+1) \\ (c,d \in \{0,1,2\}) \mapsto (c,0,+1) \\ (\Lambda,d \in \{0,1,2\}) \mapsto (\Lambda,6,-1) \\ (q,6) \mapsto (\Lambda,6,-1) \\ (\Lambda,6) \mapsto (0,5,+1) \\ \\ (\Lambda,5) \mapsto (\Lambda,5,0) \end{cases}$$

Где 0 – начальное состояние.
Действительно, если головка наткнется на последовательность aba, то машина перейдет в состояние 3, после чего просто сдвинется до ближайшего справа Λ, перейдет

в состояние 4 и пойдет влево, до ближайшего Λ, попутно «затирая» все входные данные. Когда она дойдет до Λ, то у нас на ленте будут только пробельные символы, а значит достаточно лишь вывести ответ – 1, поскольку мы в состоянии 4, а в него можно попасть, только если мы нашли *aba*. Поэтому мы ставим один и переходим в состояние 5, а также сдвигаемся на одну клетку вправо, где стоит Λ и, согласно таблице переходов, останавливаемся. Таким образом в случае нахождения последовательности символов *aba* машина работает правильно.

Если же машина не находит *aba*, то она так и идет до ближайшего справа Λ, не доходя до состояния 3 и когда она приходит в него она переходит в состояние 6 и идет до ближайшего слева Λ, попутно «затирая» входные данные, подобному случаю, когда мы все-таки нашли *aba*, только в данном случае у нее другое состояние –6. Придя в ближайшее слева Λ она выводит ответ – 0, и переходит в состояние 5, сдвигается вправо и останавливается. Таким образом в случае ненахождения последовательности символов *aba* машина также работает правильно.

Задача 4

Опишем общую идею алгоритма, который будем реализовывать с помощью МТ.
У нас сначала должны идти нули, а потом единицы, то есть по итогу у нас не должно быть подслов вида 10, но кол-во 0 и 1 не должно измениться. Значит нам нужно преобразовывать 10 в 01.
Тогда мы можем просто ходить по слову и искать самое левое 10, преобразовывать его в 01 и возвращаться в начало слова, и так до тех пор, пока 10 не останется, то есть пока мы не дойдем до ближайшего справа Λ, не найдя ни одного.
Мы можем реализовать следующим образом. Пусть s_0 – начальное состояние. Начинаем двигаться влево, сохраняя его и символы, если встречаем нули. Если же мы встречаем 1, то переходим в s_1 и сдвигаемся вправо далее, если далее встречаем 0, находясь в состоянии s_1, то переходим в s_2, пишем на его месте 1, сдвигаемся влево. Далее пишем 0, снова сдвигаемся влево и переходим в состояние s_3.
В состоянии s_3 мы должны просто двигаться влево до ближайшего *Lambda* и, дойдя до него, не изменяя его, сдвинуться вправо и перейти в состояние s_0 для которого уже описаны действия выше.
Если же мы встречаем Λ в состоянии s_1 или s_0, то мы просто останавливаемся, потому что это значит, что мы не встретили по пути 10 и слово уже отсортированно в требующемся нам порядке.
Поскольку мы смогли описать алгоритм в виде для МТ мы и можем построить такую МТ, то есть она существует.

Дискретная математика
Коллоквиум 2
Определения

12 марта 2017 г.

1. Основные определения элементарной теории вероятностей: исходы, события, вероятность события.

Вероятностным пространством – называется конечное множество U, его элементы называются **возможными исходами**.
На вероятностном пространстве задана функция $Pr : U \to [0, 1]$, такая что $\sum\limits_{x \in U} Pr[x] = 1$. Функция Pr называется вероятностным распределением, а число $Pr[x]$ называется **вероятностью исхода** $x \in U$.
Событием называется произвольное подмножество $A \subseteq U$. **Вероятностью события** A называется число $Pr[A] = \sum\limits_{x \in A} Pr[x]$.

2. Случайные графы.

Случайный граф на n вершинах – элемент вероятностного пространства U, состоящего из всех графов на n вершинах, каждому из которых приписана некоторая вероятность. В нашем курсе такой граф не содержит петель и кратных ребер, поэтому всего графов на n вершинах $2^{\binom{n}{2}} \Rightarrow |\Omega| = 2^{\binom{n}{2}}$. Понятно, что случайным будет множество ребер графа.

Пример конструкции — каждому графу Ω присвоена одинаковая вероятность, т.е. все графы равно вероятны (т.е. для любого графа $G = (V, E) \in \Omega$, для каждой пары вершин $u, v \in V$, $Pr[(u, v) \in E] = \frac{1}{2}$).

Случайные графы используются для изучения каких-то свойств графов. Например, нестрогая постановка вопроса при работе со случайными графами: велика ли вероятность того, что граф обладает данным свойством? Более конкретный пример использования: доказательство того, что при достаточно большом числе вершин, случайный граф (в равновозможной модели) будет почти всегда связен. Формально: Ω_n — вероятностное пространство состоящее из графов на n вершинах, все графы равновозможны, событие A_n — случайный граф на n вершинах связен; доказать $\lim_{n \to \infty} Pr[A_n] = 1$.

3. Условная вероятность.

Условной вероятностью события A при условии B называется число

$$Pr[A|B] = \frac{Pr[A \cap B]}{Pr[B]}$$

Заметим, что условная вероятность имеет смысл, только если $Pr[B] > 0$. Иначе знаменатель обращается в ноль.

Определение условной вероятности можно переписать следующим образом:

$$Pr[A \cap B] = Pr[B] \cdot Pr[A|B]$$

Другими словами, чтобы найти вероятность пересечения событий A и B достаточно найти вероятность события B и условную вероятность события A при условии события B.

Формула Байеса. Если вероятность событий A B положительна, то

$$Pr[A|B] = Pr[A] \cdot \frac{Pr[B|A]}{Pr[B]}.$$

Доказательство.
Следует из

$$Pr[A \cap B] = Pr[B] \cdot Pr[A|B] = Pr[A] \cdot Pr[B|A].$$

4. Независимые события. Основные свойства независимых событий.

Событие A не зависит от события B, если

$$Pr[A] = Pr[A|B]$$

Чтобы не возникало никаких тонкостей с нулевыми вероятностями полезно условиться, что вероятности событий A и B ненулевые. Из определения условной вероятности мы сразу получаем эквивалентное определение независимости событий. Событие A не зависит от события B, если

$$Pr[A \cap B] = Pr[A] \cdot Pr[B]$$

Из этой формы определения видно замечательное свойство независимости событий: она симметрична. То есть, событие A не зависит от события B тогда и только тогда, когда событие B не зависит от события A.

Отметим также, что если события A и B независимы, и вероятность события $\overline{B}$ положительна, то события A и $\overline{B}$ независимы. $\overline{B} = U \setminus B$ – событие, дополнительное к B.

События A и B, для которых $0 < Pr[B] < 1$, независимы тогда и только тогда, когда $Pr[A|B] = Pr[A|\overline{B}]$.

5. Случайная величина и математическое ожидание.

Случайная величина – это числовая функция на вероятностном пространстве, то есть функция вида $f : U \to R$. То есть, по сути, случайная величина – это обычная числовая функция, но теперь на её аргументах задано вероятностное распределение. Таким образом, например, мы можем говорить о вероятности того, что случайная величина f равна какому-то конкретному значению a: это есть просто вероятность события

$\{u \in U \mid f(u) = a\}$. Случайные величины представляют собой числовые характеристики вероятностных экспериментов, и на самом деле, мы с ними уже неоднократно сталкивались, просто не говорили об этом. Например, если мы бросаем кубик, то исходом эксперимента является выпадение той или иной грани, а случайной величиной – число написанное на грани (каждой грани соответствует своё число – это функция).

Пусть вероятностное событие состоит из k исходов, случайная величина $f : U \to R$ принимает на них значения $a_1, \ldots, a_k$ соответственно и вероятности исходов равны $p_1, \ldots, p_k$ соответственно. В частности, $\sum\limits_{i=1}^{k} p_i = 1$. Предположим, что мы повторяем эксперимент по выбору случайного элемента из U n раз. Если n достаточно большое, то случайная величина f примет значение a_1 примерно $p_1 n$ раз, значение a_2 – примерно $p_2 n$ раз, и так далее, значение a_k – примерно $p_k n$ раз. Подсчитаем теперь примерное среднее арифметическое значений случайной величины f в этих экспериментах:

$$\frac{a_1 p_1 n + a_2 p_2 n + \ldots + a_k p_k n}{n} = \sum_{i=1}^{k} a_i p_i$$

Математическим ожиданием случайной величины f, принимающей значения $a_1, \ldots, a_k$ с вероятностями $p_1, \ldots, p_k$ соответственно, называется величина

$$E[f] = \sum_{i=1}^{k} a_i p_i$$

Математическое ожидание линейно.

Неравенство Маркова. Пусть f – случайная величина, принимающая только неотрицательные значения. Тогда для всякого $\alpha > 0$ верно

$$Pr[A|B] = Pr[A] \cdot \frac{Pr[B|A]}{Pr[B]}.$$

То есть вероятность того, что случайная величина f сильно больше своего математического ожидания невелика. Заметим, что лемма становится содержательной, только когда $\alpha > E[f]$.

*?. Определение бесконечного множества.

Определение: Множество A – конечно тогда и только тогда, когда $\exists n \in \mathbb{N}_0 : A \sim [n]$ ($[n] = \{1, 2, \ldots, n\}, [0] = \varnothing$).

Запомните, что: $n > k \Rightarrow [n] \nsim [k]$.

Определение: Множество бесконечно тогда и только тогда, когда оно не конечно.

6. Определение равномощных множеств. Основные свойства равномощности.

Определение: Равномощными множествами называются такие множества, между которыми установима биекция. Обозначение: $A \sim B$.

Очевидные свойства равномощных множеств: $\forall A$ – множеств.

- $A \sim A$.
- $A \sim B \Rightarrow B \sim A$.
- $A \sim B, B \sim C \Rightarrow A \sim C$.

7. Определение счетного множества. Примеры.

A – бесконечно. Значит A – не пусто. $\exists a_o \in A$. Пусто ли $A \setminus a_0$? Нет. Иначе A – содержит один элемент и конечно.

Тогда $\exists a_1 \in A \setminus a_0$. Множество и без этих двух элементов бесконечно. Ну и так далее.

Определение: Получившееся множество $A' = \{a_0, a_1, a_2, \ldots, a_n, \ldots\}$ назовем счётным (равномощным множеству натуральных чисел). Биекция в этом случае очевидна: $f : i \mapsto a_i$.

Утверждение: $\mathbb{N}$ – бесконечно.

Доказательство. Пусть это не так и $\exists f : [n] \to \mathbb{N}$ – инъекция. Тогда верно следующее: $\mathbb{N} \ni max\{f(0), f(1), \ldots, f(n)\} + 1 \notin f([n])$. А значит f – не биекция. А значит $\mathbb{N}$ не равномощно никакому $[n]$. **Q.E.D.**

Примеры счётных множеств:

- $\{0,1\}^*$ – множество двоичных слов.
- $\mathbb{N}$ – множество натуральных чисел (целые положительные и 0).
- $\mathbb{N} \times \mathbb{N}$ – множество пар натуральных чисел.
- $\mathbb{N}^*$ – множество конечных последовательностей натуральных чисел.

Утверждение: Множество бесконечно тогда и только тогда, когда оно равномощно какому-то своему подмножеству.

Доказательство. Докажем, что если множество бесконечно, то оно равномощно некоторому подмножеству.

Как мы уже выяснили, в любом бесконечном множестве есть счётное подмножество. Пусть $B = \{b_0, b_1, \ldots, b_n, \ldots\}$ – счётное подмножество бесконечного множества A.

Установим биекцию $f : A \setminus \{b_0\} \to A$.

$$f(x) = \begin{cases} b_{n-1}, x \in B \\ x, x \notin B \end{cases}$$

Получили то, что и требовалось.

В обратную сторону доказывается на семинарах, но примерно так: пусть $B \subset A, B \sim A$. Пусть A – конечно. Тогда $|B| < |A|$ **Q.E.D.**

8. Основные свойства счетных множеств.

1. A – счётное множество. Тогда $A' \subseteq A$ счётно или конечно.

   Доказательство. $A = \{a_0, a_1, \ldots, a_n, \ldots\}$. Вычеркнем все элементы, в A' не входящие. $A' = \{a_{j_0}, a_{j_1}, \ldots, a_{j_n}, \ldots\}$.

   Если последовательность $\{a_{j_n}\}$ конечна, то и A' конечно. Если она бесконечна, то A' очевидно счётно. **Q.E.D.**

2. Если A,B – счётные, то и $A \cup B$ счётно.

   Доказательство. $A = (a_0, a_1, \ldots, a_n, \ldots)$. $B = (b_1, b_2, \ldots, b_n, \ldots)$.

   $A \cup B = (a_0, b_0, a_1, b_1 \ldots, a_n, b_n, \ldots)$.

   Но может получиться так, что в новой последовательности некоторые элементы встречаются по два раза (они входят в оба множества). Вычеркнем каждый такой элемент по одному разу. И получим последовательность, задающую счётное множество. **Q.E.D.**

3. $\mathbb{Z}$ – счётно.

   Доказательство. $Z = \mathbb{N} \cup (-\mathbb{N})$ – объединение счётных множеств. Счётно по свойству 2 ($-A = \{-a \mid a \in A\}$). **Q.E.D.**

4. Если A – счётно, а B – конечно или счётно, то $A \cup B$ счётно.

   Доказательство. Доказывается аналогично свойству 2. **Q.E.D.**

5. Если A – счётно. И $B_1, B_2, \ldots, B_k$ – счётны или конечны, то $A \cup B_1 \cup \ldots \cup B_k$ – счётно.

   Доказательство. К доказательству свойства 4 нужно добавить доказательство по индукции. **Q.E.D.**

6. Счётное объединение конечных или счётных множеств конечно или счётно.

   $\{A_0, A_1, \ldots, A_n, \ldots\} = \mathfrak{F} \sim \mathbb{N}$. A_i – множество. $\mathfrak{F}$ называется семейством множеств. $A = \bigcup\limits_{i=0}^{\infty} A_i$.

   Утверждение: A – счётно.

   Доказательство.

   $$A_0 = (a_{00}, a_{01}, \ldots, a_{0n}, \ldots)$$
   $$A_1 = (a_{10}, a_{11}, \ldots, a_{1n}, \ldots)$$

   Некоторые из множеств могут быть конечны. Дополним их до счётных пустым символом $\lambda \notin A$.

Построим последовательность: $a_{00}, a_{01}, a_{10}, a_{02}, a_{11}, a_{20}, \ldots$. (то есть проходим последовательно все значения сумм индексов от 0 до ∞).

Теперь исключим из последовательности повторения и символы λ. Получим требуемую последовательность $(a'_0, a'_1, \ldots, a'_n, \ldots)$.

Теперь получим функцию $f : [n] \to A$ или $f : \mathbb{N} \to A$. f – биекция. В первом случае множество конечно, во втором счётно.

Можно было бы и не вводить λ, а исключать эти элементы сразу, но так проще (нет никаких условий). **Q.E.D.**

Примеры:

- Пусть $A_i = \{i\}$. Тогда $A = \mathbb{N}$ (счётно).
- Пусть $A_i = \{1\}$. Тогда $A = \{1\}$ (конечно).

7. Декартово произведение счётных множеств счётно. Напомним, что

$$A \times B = \{(a; b) \mid a \in A, b \in B\}$$

Доказательство. По определению декартово произведение есть множество всех упорядоченных пар вида $\langle a, b\rangle$, в которых $a \in A$ и $b \in B$. Разделим пары на группы, объединив пары с одинаковой первой компонентой (каждая группа имеетвид $\{a\} \times B$ для какого-то $a \in A$). Тогда каждая группа счётна, поскольку находится во взаимно однозначном соответствии с B (пара определяется своим вторым элементом), и групп столько же, сколько элементов в A, то есть счётное число. **Q.E.D.**

8. Если A – счётно, то A^k – счётно.

Доказательство. Очевидно по индукции из свойства 7. **Q.E.D.**

9. $\mathbb{Q}$ – счётно.

Доказательство. Рассмотрим множество $\mathbb{Q}_p$ несократимых дробей. Пусть функция $f : \mathbb{Q}_p \to \mathbb{Z} \times \mathbb{N}_+$ – инъекция (она переводит дробь в пару чисел числитель-знаменатель). Тогда она является биекцией на $f(\mathbb{Q}_p) \subset \mathbb{Z} \times \mathbb{N}_+$. Причём $f(\mathbb{Q}_p)$ тогда счётно по свойству 1 так как не является конечным, а $\mathbb{Z} \times \mathbb{N}_+$ счётно по свойству 7. **Q.E.D.**

10. Пусть A^* – конечные последовательности конечного (непустого) или счётного алфафита A.

Утверждение: A^* – счётно

Доказательство. $A^* = \bigcup\limits_{n=0}^{\infty} A^n$. При этом A^n – слова длины n. A^n – счётно по свойству 8. И тогда само A^* счётно по свойству 6. **Q.E.D.**

11. Определение: $\alpha \in \mathbb{R}$ – алгебраическое число тогда и только тогда, когда α – корень некоторого многочлена с целыми коэфициентами.

Утверждение: Множество алгебраических чисел счётно.

Доказательство. Приведём только план доказательства:

(a) Докажем, что многочленов степени n ($n \in \mathbb{N}$) с целыми коэфициентами счётно.

(b) Для каждого из этих многочленов есть не более n корней – алгебраических чисел.

(c) Удаляем повторяющиеся корни.

(d) Получим все алгебраические числа, которых, очевидно, счётно.

Q.E.D.

9. Определение множества мощности континуум. Примеры.

Определим действительные числа следующим образом: сопоставим каждому $x \in \mathbb{R}$ двоичное число: $\pm\underbrace{10110\ldots1011}_{\text{целая часть}}.\overbrace{110001\ldots00110}^{\text{дробная часть}}$. Считаем известным, что ряд из каких-то степеней двоек сходится, причём запрещаем в числах данного вида "хвосты из единиц".

Определение: Будем говорить, что множество X имеет мощность континуум, если $X \sim \mathbb{R}$.

Примеры:

- $\Phi(\mathbb{N})$ – множество всех подмножеств $\mathbb{N}$.
- $2^{\mathbb{N}}$ – множество последовательностей натуральных чисел.
- $\mathbb{R}$ – само множество действительных чисел.

10. Основные свойства континуальных множеств.

1. Любое континуальное множество имеет счётное подмножество.
2. Мощность объединения не более чем континуального количества множеств, каждое из которых не более чем континуально, не превосходит континуума.
3. При разбиении континуального множества на конечное или счётное число частей хотя бы одна из частей будет иметь мощность континуум.

11. Булевы функции. Задание булевых функций таблицами истинности.

Булева функция от n аргументов - отображение из B^n в B, где B - {0,1}. Количество всех n-арных булевых функций равно 2^{2^n}. Булеву функцию можно задать таблицей истинности, поскольку она задается конечным набором значений.

12. Определение полного базиса. Примеры полных и неполных базисов.

Полный базис: Базис B – *полный*, если любую булеву функцию можно вычислить схемой в базисе B.

Примеры полных базисов:

1. Стандартный Базис
2. Дизъюнкция, инверсия
3. Конъюнкция, инверсия
4. Импликация, инверсия
5. Базис Жегалкина $\{1, \oplus, \vee\}$
6. Стрелка Пирса
   (a) $0 \downarrow 0 = 1, 0 \downarrow 1 = 0, 1 \downarrow 0 = 0, 1 \downarrow 1 = 0$
   (b) $X \downarrow X \equiv \neg X$ — отрицание
   (c) $(X \downarrow X) \downarrow (Y \downarrow Y) \equiv X \wedge Y$ — конъюнкция
   (d) $(X \downarrow Y) \downarrow (X \downarrow Y) \equiv X \vee Y$ — дизъюнкция
   (e) $((X \downarrow X) \downarrow Y) \downarrow ((X \downarrow X) \downarrow Y) = X \to Y$ — импликация
7. Штрих Шеффера
   (a) $0 \| 0 = 1, 0 | 1 = 1, 1 | 0 = 1, 1 | 1 = 0$
   (b) $X \,|\, X = \neg X$ — отрицание
   (c) $(X \,|\, X) \,|\, (Y \,|\, Y) = X \vee Y$ — дизъюнкция
   (d) $(X \,|\, Y) \,|\, (X \,|\, Y) = (X \wedge Y)$ — конъюнкция
   (e) $X \,|\, \neg X$ — константа 1

Примеры неполных базисов:

1. Монотонный базис
2. $\{\wedge, \oplus\}$
3. $\{1, \wedge\}$
4. $\{1, \oplus\}$
5. Конъюнкция, дизъюнкция и разность
6. Большинство одноэлементных базисов

13. Разложение Рида.

Разложением Шеннона функции $f : \{0,1\}^n \to \{0,1\}$ по переменной x_i называется представление фукнции f в виде:

$$f(x_n, \dots, x_i, \dots, x_1) = \overline{x_i} \cdot f(x_n, \dots, 0, \dots x_1) \vee x_i \cdot f(x_n, \dots, 1, \dots, x_1)$$

Разложением Рида называется следующее представление функции:

$$f(x_n, \dots, x_i, \dots, x_1) = g_0 \oplus (g_0 \oplus g_1) \cdot x_i),$$

$$g_0 = f(x_n, \dots, 0, \dots x_1)$$
$$g_1 = f(x_n, \dots, 1, \dots, x_1)$$

14. ДНФ, СДНФ и СКНФ

Необходимое определение. Простой конъюнкцией называется конъюнкция одной или нескольких переменных или их отрицаний, причём каждая переменная встречается не более одного раза.

Простая конъюнкция

- **полная**, если в неё каждая переменная (или её отрицание) входит ровно 1 раз;
- **монотонная**, если она не содержит отрицаний переменных.

Дизъюнктивная нормальная форма, она же ДНФ – нормальная форма, в которой булева функция имеет вид дизъюнкции нескольких *простых* конъюнктов.

Пример ДНФ: $f(x,y,z) = (x \wedge y) \vee (y \wedge \neg z)$.

Совершенная дизъюнктивная нормальная форма, СДНФ – ДНФ, удовлетворяющая условиям:

- в ней нет одинаковых простых конъюнкций,
- каждая простая конъюнкция полная.

Пример СДНФ: $f(x,y,z) = (x \wedge \neg y \wedge z) \vee (x \wedge y \wedge \neg z)$.

Конъюктивная нормальная форма и Совершенная конъюктивная нормальная форма определяются аналогично:

- Конъюктивная нормальная форма – конъюнкция простых дизъюнктов
- СКНФ – КНФ, в которой каждый дизъюнкт — полный.

15. Полином Жегалкина

Полином Жегалкина — полином с коэффициентами вида 0 и 1, где в качестве произведения берётся конъюнкция, а в качестве сложения исключающее или. Каждая булева функция единственным образом представляется в виде полинома Жегалкина.

16. Определение схемы в некотором функциональном базисе. Представление схем графами.

Схема — это функция, заданная последовательностью присваиваний.

Иными словами, булевой схемой от переменных $x_1, \ldots, x_n$ в некотором функциональном базисе мы будем называть последовательность булевых функций $g_1, \ldots, g_s$, в которой всякая g_i или равна одной из переменных, или получается из предыдущих применением одной из функций из базиса.

Также в профессиональной среде схемы называют SLP (*straight line programmes*).

Рассмотрим такую функцию f, определенную для булевых значений (*булеву функцию*): $f : \{0,1\}^n \to \{0,1\}$.

Базисом B булевой функции будем называть некий набор $B : \{f_1, f_2, \ldots, f_n\}$, где $f_1 \ldots f_n$ - булевы функции.

Булева схема в базисе B — последовательность функций $x_1, x_2, x_3 ... x_n, x_{n+1} := S_1, x_L := S_{L-n}$, которая вычисляет $x_L(x_1, \ldots, x_n)$.

$$S_j = g(S_{i_1}, ..., s_{i_r}), g \in B, i_< j$$

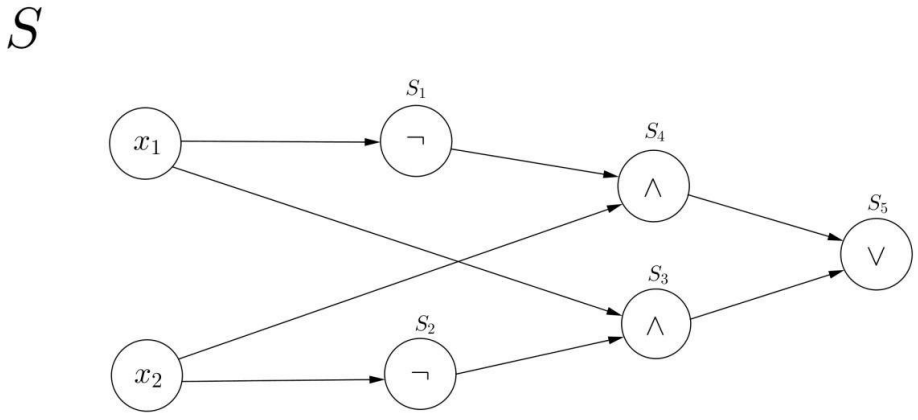
Стандартный базис есть базис, состоящий из операций отрицания, конъюнкции и дизъюнкции: $\{\neg, \vee, \wedge\}$

Схемы можно представлять в виде графов:

Пример 1

Зададим булеву схему с помощью стандартного базиса.

$x_1, \ldots, x_n, s_1 := \neg x_1, s_2 = \neg x_2, s_3 := x_1 \wedge s_2, s_4 := x_2 \wedge s_1; s_5 := s_3 \vee s_4$



Если $x_2 = 0$, то $s_5 = x_1$

Если $x_2 = 1$, то $s_5 = \neg x_1$

Результатом выполнения булевой схемы является сложение по модулю 2 (1, если значения x_1 и x_2 разные) - $\oplus$.

Пример 2

Составим схему, которая является сложением по модулю 2 n переменных. Приведём индуктивное доказательство её существования:

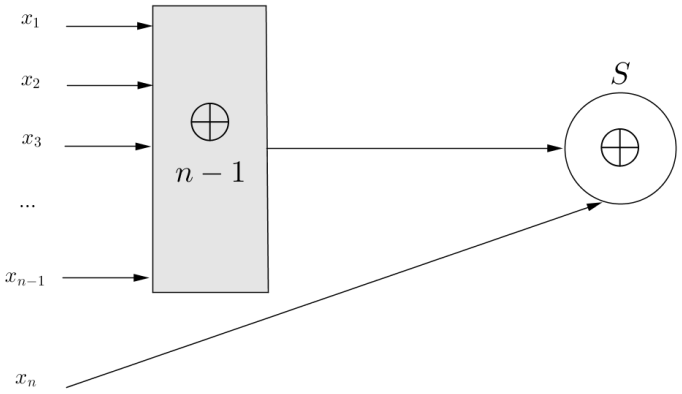
1. База индукции — $n = 2$. Сложение 2 переменных по модулю 2 возможно по схеме, описанной выше

2. Предположим существование такой схемы для $n - 1$ переменных

$$x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1}$$

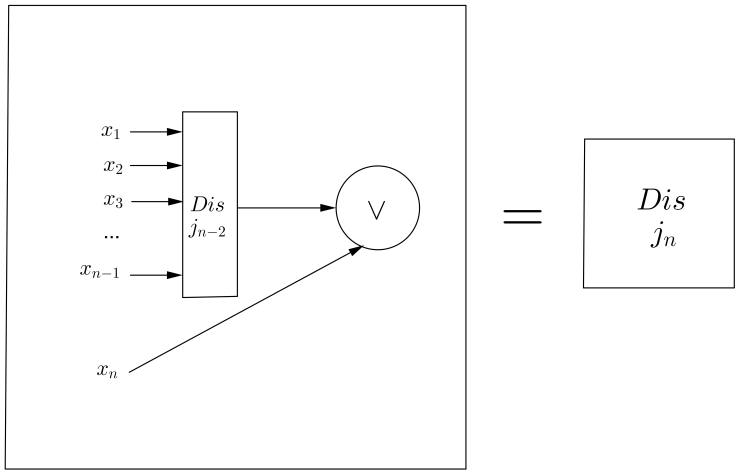
3. Рассмотрим сложение по модулю 2 n переменных. Представим его как сложение по модулю 2 x_n с результатом предыдущего шага. Существование второго слагаемого объясняется предположением индукции. Сложение с x_n можно выполнить по схеме выше.

$$x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1} \oplus x_n = (x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1}) \oplus x_n$$



Пример 3

Дизъюнкция n переменных — аналогично, по индукции. Такие рассуждения можно построить и для конъюнкции.



17. Определение схемной сложности функции.

Размер булевой схемы — это количество присваиваний в схеме $g_1, \ldots, g_L$ для вычисления функции $f : \{0,1\}^n \to \{0,1\}$.

Сложность функции f в базисе B — это минимальный размер булевой схемы, вычисляющей функцию f в базисе B. Если базис не указывают – имеют в виду стандартный базис $\{\neg, \vee, \wedge\}$. *Обозначение:* $C(f)$.

Определения различных оценок сложностей аналогичны тем, что мы используем на алгоритмах и в математическом анализе:

$$o(f(n)) = \{g(n) \mid \forall\ c > 0\ \exists\ n_0 > 0\ : \forall n \geq n_0 \Rightarrow 0 \leq g(n) < c \cdot f(n)\}$$

$$O(f(n)) = \{g(n) \mid \exists\ c > 0,\ \exists\ n_0 > 0\ : \forall n \geq n_0 \Rightarrow 0 \leq g(n) \leq c \cdot f(n)\}$$

$$\omega(f(n)) = \{g(n) \mid \forall\ c > 0\ \exists\ n_0 > 0\ : \forall n \geq n_0 \Rightarrow 0 \leq c \cdot f(n) < g(n)\}$$

$$\Omega(f(n)) = \{g(n) \mid \exists\ c > 0,\ \exists\ n_0 > 0\ : \forall n \geq n_0 \Rightarrow 0 \leq c \cdot f(n) \leq g(n)\}$$

$$\Theta(f(n)) = \{g(n) \mid \exists\ c_1 > 0, \exists\ c_2 > 0, \exists\ n_0 \in \mathbb{N} :\ \forall\ n \geq n_0 \Rightarrow c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n)\}$$

18. Основные свойства вычислимых функций.

Определение: Функция называется вычислимой, если для неё существует некоторый алгоритм вычисления.

Первое свойство алгоритмов. *Композиция вычислимых функций вычислима.*

Доказательство. Пусть существуют вычислимые функции f, переводящая множество входов X в множество выходов Y и g, переводящая Y в Z.

$$\begin{cases} f : X \to Y \\ g : Y \to Z \end{cases}$$

Построить $g \circ f : X \to Z$ достаточно просто. На первом шаге нужно применить функцию f. Далее берём множество выходов f и подаём на вход в g. На выходе получим некоторое множество выходов Z. Вычислимая композиция построена.

Algorithm 1 Алгоритм получения композиции функций

```
1: function Composition(x)
2:     t ← f(x)
3:     return g(t)
4: end function
```

Q.E.D.

Итак, есть вычислимая биекция $\pi : \mathbb{N}^* \to \mathbb{N}$.

Построим композицию $\mathbb{N}^* \to \mathbb{N} \to \{0, 1\}$.

$$\begin{cases} \pi^{-1} : B \to A \\ f : A \to A \\ \pi : A \to B \end{cases}$$

Построить это можно применением композиции $\pi \circ f \circ \pi^{-1}$. Получим алгоритм из B в B. Поэтому нам, по сути, **без разницы какие множества на входе и выходе** так как можно получить легко из одного другое.

Заметим, что если некоторая биекция π вычислима, то обратная ей π^{-1} также будет вычислима. Способ вычисления схож с алгоритмом для диофантовых уравнений.

Рассматриваем все возможные значения входов и вычисляем $\pi^{-1}(x)$.

Algorithm 2 Алгоритм построения обратной функции для биекции

```
1: function RevBiection(x)
2:     for n := 0 ... ∞ do
3:         if π(n) = x then
4:             return n
5:         end if
6:     end for
7: end function
```

Стоит отметить, что данный алгоритм всегда завершается так как (по определению) биекция сюръективна и найдется номер n для которого $\pi(n) = x$.

19. Определение разрешимого множества.

Вспомним что такое характеристическая функция множества S:

$$\chi_S(x) = \begin{cases} 1, \, x \in S \\ 0, \, x \notin S \end{cases}$$

Определение: Множество называется разрешимым, если его характеристическая функция вычислима.

20. Определение перечислимого множества.

Определение: Алгоритм перечисления – такой алгоритм, у которого нет входа, он работает и может выводить некоторые числа, причём все напечатанные числа составляют счетное множество.

Определение 1: Множество S называется перечислимым, если есть алгоритм перечисления всех его элементов.

Определение 2: Множество S – перечислимо, если существует такая вычислимая функция:

$$f : \mathbb{N} \to \mathbb{N} \begin{cases} f(\mathbb{N}) = S \\ \text{Область определения } f \text{ равна либо } \mathbb{N}, \text{либо } [n]. \end{cases}$$

Теорема. *Определения* 1 *и* 2 *эквивалентны.*

Доказательство.

$\Longrightarrow$ Пусть A – алгоритм перечисления множества S. Тогда возьмём следующий алгоритм B: принимает на вход число n, запускает алгоритм A и считает, сколько чисел напечатано. Как только вывели $n+1$ слово – алгоритм печатает результат.

Покажем, что соблюдаются свойства вычислимой функции:

1. $B(n) = S$, так как $\forall\ n\ \exists\ B(n) \Rightarrow \begin{cases} \forall\ x \in S\ \exists\ B(x) \\ \forall\ x \notin S \text{ никогда не выведет } B(x) \end{cases}$
2. Пусть S – бесконечное множество, тогда функция, задаваемая B, определена везде, значит $dom\ B = \mathbb{N}$. Если B работает на n числах, то алгоритм переберёт их и остановится.

$\Longleftarrow$ Возьмём следующий алгоритм перечисления B для множества S:

Algorithm 3 Алгоритм перечисления разрешимого множества

```
1: function PrintSet(S)
2:     for i := 0 ... ∞ do
3:         if f(i) = 1 then
4:             print i
5:         end if
6:     end for
7: end function
```

Если функция определена для некоторых n чисел, то ровно их он и напечатает. Если $dom\ f = \mathbb{N}$, то алгоритм никогда не остановится, то есть напечатает всю область определения f. Значит, существует алгоритм, перечисляющий S.

Q.E.D.

21. Свойства перечислимых множеств.

Утверждение: Если множество S разрешимо, то оно перечислимо.

Доказательство. Алгоритм перечисления множества S использует алгоритм раз- решения множества S. Он перебирает все числа, начиная с 0; для каждого числа n вычисляет индикаторную функцию $\chi S(n)$ и печатает число n, если полученное значение равно 1.

Algorithm 4 Алгоритм перечисления множества S

```
1: function Print(S(n))
2:     for n := 0 ... ∞ do
3:         if χ_S(n) = 1 then
4:             print n
5:         end if
6:     end for
7: end function
```

Корректность такого алгоритма ясна из определений. **Q.E.D.**

Пусть S – перечислимое непустое множество. Тогда для S выполнены следующие свойства:

1. S – область определения вычислимой функции.
2. S – область значений вычислимой функции.
3. S – область значений всюду определённой вычислимой функции.

Пусть A и B – перечислимые множества. Тогда:

1. $A \cup B$ перечислимо.
2. $A \cap B$ перечислимо.

Утверждение: Существует перечислимое неразрешимое множество.

Доказательство. Рассмотрим вычислимую функцию $f(x)$, не имеющую всюду определённого вычислимого продолжения. Её область определения F будет искомым множеством. В самом деле, F перечислимо (по одному из определений перечислимости). Если бы F было разрешимо, то функция

$$g(x) = \begin{cases} f(x), \text{если } x \in F \\ 0, \text{если } x \notin F. \end{cases}$$

была бы вычислимым всюду определённым продолжением функции f (при вычислении $g(x)$ мы сначала проверяем, лежит ли x в F, если лежит, то вычисляем $f(x)$). **Q.E.D.**

22. Определение универсальной вычислимой функции.

Определение: Универсальная вычислимая функция:

$$U : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \mid \forall f\text{-вычислимая } \exists\, p : f(x) = U(p,\, x).$$

Также полезно знать про отладочную функцию:

Определение: Отладочная функция:

$$F : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \{0,\, 1\} \text{ – всюду определённая, причём:}$$

$$\begin{cases} \text{При фиксированных } p \text{ и } x \text{ монотонна по } t \\ F(p,\, x,\, t) = 0 \Leftrightarrow U(p,\, x) \text{ не определена} \\ F(p,\, x,\, t) = 1 \Leftrightarrow \text{ программа } p \text{ на входе } x \text{ заканчивает работу за } t \text{ шагов.} \end{cases}$$

23. Определение отладочной функции.

Пусть $U(p, x)$ – универсальная вычислимая функция. Для данной у.в.ф. существует функция $F : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \{0, 1\}$, называемая отладочной, для которой выполняются следущие свойства:

- $F(p, x, t)$ не убывает по t (Т.е. $\forall x, p \in N : t_0 < t_1 \Leftrightarrow F(p, x, t_0) \leq F(p, x, t_1)$)
- $U(p, x)$ не определена $\Leftrightarrow \forall t \in N : F(p, x) = 0$

Неформально говоря, значение функции $F(p, x, t)$ равно 0 тогда и только тогда, когда программа p на входе x не закончила работу за количество шагов t. В противном случае значение функции $F(p, x, t)$ равно 1.

24. Определение главной универсальной вычислимой функции.

Определение: Главная универсальная функция (гёделева) – такая универсальная функция, что для любой вычислимой функции $V(n,\ x)$ существует всюду определённая вычислимая функция $s(n)$, что:

$$\forall\ n,\ x \Rightarrow V(n,\ x) = U(s(n),\ x).$$

Неформально это значит, что главная универсальная функция позволяет транслировать в себя любую другую унивесальная функция. Ну, вот например, есть язык C++, его можно назвать главной универсальной функцией так как любую программу на другом универсальном языке можно переписать на C++ автоматически (при помощи *транслятора*).

25. Формулировка теоремы Успенского–Райса.

Пусть есть некоторое свойство, которое мы хотим проверить для некоторой функции.

Формально: Пусть $\{f : \mathbb{N} \to \mathbb{N}\}$ – множество вычислимых функций. Разделим его на два непересеающихся подмножества A и $\overline{A}$.

$$\{f \mid f : \mathbb{N} \to \mathbb{N}\} = A \cup \overline{A}$$

A – множество тех функций, для которых выполняется некое свойство, $\overline{A}$ – множество тех функций, для которых это свойство не выполняется.

Возьмём некоторую универсальню функцию $U(p,\ x)$.

Обозначим за P_A множество всех p таких, что $U(p,\ x) \in A$.

$$P_A = \{p \mid U(p, x) \in A\}$$

Тогда вопрос можно поставить так: разрешимо ли множество программ, удовлетворяющих нашему свойству? На этот вопрос и отвечает теорема Успенского-Райса:

Теорема Успенского-Райса. *Если A – нетривиально ($A \neq \oslash$, $\overline{A} \neq \oslash$), а $U(q,\, x)$ – главная универсальная функция, то множество P_A неразрешимо.*

Введём для удобства ещё функции $\varepsilon \in \overline{A}$ (нигде не определённая) и $\xi \in A$ (какая-то функция, удовлетворяющая условию). Сделать это можно по аксиоме выбора.

Если A – это множество нигде не определённых функций, то поменяем их местами так как P_A разрешимо тогда и только тогда, когда его дополнение разрешимо.

26. Формулировка теоремы о неподвижной точке.

Теорема о неподвижной точке. *Пусть U – главная универсальная функция, $h(n)$ – любая всюду определённая вычислимая функция. Тогда:*

$$\exists\ q\ :\ U(q,\, x) = U(h(q),\, x).$$

Честно сказать, не все учёные понимают эту теорему, однако её можно объяснить неформально так: для любой программы на любом универсальном языке существует ещё одна программа, которая делает то же самое (то есть программы совпадают).

27.1. Определение машины Тьюринга (с одной лентой).

Мы рассмотрим классическую модель вычислений, на которой будут основано точное определение вычислимых функций, – машины Тьюринга (МТ).

МТ состоит из

- бесконечной в две стороны ленты, в ячейках которой могут быть записаны символы алфавита A (некоторого конечного множества)
- головки, которая может двигаться вдоль ленты, обозревая в каждый данный момент времени одну из ячеек
- оперативной памяти, которая имеет конечный размер (другими словами, состояние оперативной памяти – это элемент некоторого конечного множества, которое называется множеством состояний МТ Q)
- таблицы переходов (или программы), которая задаёт функцию $\delta : A \times Q \to A \times Q \times \{-1,\, 0,\, +1\}$

Поскольку таблица переходов – это функция на конечном множестве, её возможно задать таблицей. Каждая строка таблицы – это пять значений a, q, $a^{'}$, $q^{'}$, d, (другие способы записи: $\delta(a,\, q) = (a^{'},\, q^{'},\, d)$ или $\delta : (a,\, q) \mapsto (a^{'},\, q^{'},\, d)$), которые описывают следующий порядок действий МТ: если головка МТ находится над ячейкой, содержащей символ a, а состояние МТ равно q, то на очередном такте работы МТ записывает в текущую ячейку символ $a^{'}$, изменяет состояние на $q^{'}$ и сдвигает головку на d ячеек (отрицательное значение отвечает сдвигу влево, положительное – сдвигу вправо, 0 – не сдвигается).

Работа МТ состоит из последовательного выполнения тактов в соответствии с таблицей переходов. Может так случиться, что для текущей пары значений $(a,\, q)$ функция

переходов не определена. В этом случае работа машины заканчивается (машина останавливается). Обычно среди состояний МТ выделяют множество Q_f финальных состояний – таких состояний q_f, что таблица переходов не определена для всех пар (a, q_f). Попав в финальное состояние, машина обязательно остановится, отсюда и название.

Лента МТ бесконечна и это не соответствует нашей интуиции об алгоритмах: алгоритм на каждом шаге работы оперирует лишь данными конечного размера. Чтобы учесть это обстоятельство, мы предполагаем, что в алфавите машины есть специальный символ Λ (пробел или пустой символ) и все ячейки ленты за исключением конечного числа содержат пустые символы. Это свойство ленты сохраняется при работе МТ, поскольку за такт работы меняется содержимое не более одной ячейки ленты.

Состояние такой машины уже описывается конечными данными. Мы будем использовать конфигурации. Конфигурация – это слово в алфавите $A \cup Q$, в котором первый и последний символы непустые, и ровно один символ принадлежит множеству состояний. Договоримся считать, что символ состояния записывается слева от символа в той ячейке, над которой находится головка МТ. На ленте слева и справа от символов конфигурации стоят только пустые символы.

У МТ есть 2 основных конфигурации: начальная q_0, из которой МТ начинает работу, и финальная q_f.

Конфигурации МТ преобразуются такт за тактом, порождая последовательность конфигураций $c_0 = q_0 u$, c_1, $c_2, \ldots, c_t, \ \ldots$

Эта последовательность бесконечна, если машина не останавливается, и конечна в противном случае. Результатом работы является та часть финальной конфигурации, которая расположена между символом состояния и ближайшим к нему пустым символом справа.

27.2. Определение машины Тьюринга (с несколькими лентами).

Определение: Машина, у которой не одна лента, а несколько (фиксированное число для конкретной машины), называется *многоленточной*.

На каждой ленте есть своя головка. За такт работы головки могут перемещаться по всем лентам. Действие на такте работы зависит как от состояния машины, так и от всего набора символов, которые видят головки машины на всех лентах.

Чтобы задать машину с h лентами, нужно указать:

- алфавит A, в котором выделен пустой символ Λ
- множество состояний Q, в котором выделено начальное состояние q_0
- таблицу переходов, которая теперь является функцией вида $\delta : A^h \times Q \to A^h \times Q \times \{-1, 0, +1\}^h$ (первый аргумент – символы, которые машина видит на ленте, последний – команды движения для головок на каждой ленте).
- выделить среди лент ленту входа и ленту результата (возможно, что это одна и та же лента)

Таблица переходов по прежнему является функцией на конечном множестве, поэтому её возможно задать таблицей. Работа МТ состоит из последовательного выполнения тактов в соответствии с таблицей переходов. Может так случиться, что для текущего набора

значений $(a_1, a_2, \ldots, a_h, q)$ функция переходов не определена. В этом случае работа машины заканчивается (машина останавливается). Как и раньше, можно ввести множество финальных состояний Q_f, т.е. тех состояний q_f, для которых таблица переходов не определена для всех значений $(a_1, a_2, \ldots, a_h, q_f)$. В финальном состоянии машина обязательно останавливается.

Мы предполагаем, что h-МТ начинает работу в состоянии q_0, а все ленты кроме ленты входа содержат только пустые символы. На ленте входа записано входное слово, и головка находится над первой слева ячейкой, содержащей символы этого слова. Поскольку за такт работы меняется содержимое не более одной ячейки ленты, в процессе работы машины на каждой ленте будет записано лишь конечное количество непустых символов.

Конфигурация многоленточной машины может быть задана набором конфигураций на каждой ленте $(u_1 q_{v_1}, u_2 q_{v_2}, \ldots, u_h q_{v_h})$.

Символ состояния один и тот же, так как по нашим определениям состояние есть у машины, а не у головки.

Далее нам будет удобен другой способ представления конфигурации машины. Выровняем ленты и будем рассматривать *окно*, в которое заведомо помещаются все непустые символы на каждой ленте. В таком случае конфигурация однозначно определяется матрицей размера $h \times N$, в которой записаны символы на лентах. Нужно ещё указать положения головок на лентах (они-то не обязательно выровнены – машина способна перемещать головки независимо). По этой причине будем помещать в матрицу не символы алфавита A, а пары $(a, \hat{q})$, где $\hat{q}$ указывает, расположена ли на данной ленте головка над данной ячейкой. Если да, то $\hat{q} \in Q$ – текущее состояние машины. Если нет, то $\hat{q}$ – какой-то символ не из Q, который указывает, что над над данной ячейкой на данной ленте нет головки. Будем для единообразия использовать в качестве такого символа Λ. Такую матрицу в дальнейшем называем матрицей конфигурации. Вот пример матрицы начальной конфигурации для двухленточной машины:

(Λ, q_0)	(Λ, Λ)	(Λ, Λ)	(Λ, Λ)	(Λ, Λ)
(a, q_0)	(b, Λ)	(a, Λ)	(a, Λ)	(b, Λ)

Как и для одноленточной машины, работа h-МТ порождает последовательность конфигураций $c_0 = q_0 u$, c_1, $c_2, \ldots, c_t, \ldots$

Эта последовательность бесконечна, если машина не останавливается, и конечна в противном случае. Результатом работы является та часть финальной конфигурации на ленте результата, которая расположена между положением головки и ближайшим к нему пустым символом справа. Например, если у двухленточной МТ лента результата – нижняя, то результатом работы МТ, остановившейся в конфигурации, заданной окном

(Λ, Λ)	(Λ, Λ)	(a, q_f)	(a, Λ)	(Λ, Λ)
(a, Λ)	(b, q_f)	(a, Λ)	(Λ, Λ)	(b, Λ)

будет ba.

28. Определение функции, вычислимой на машине Тьюринга.

Определение: МТ M (h-МТ, если многоленточная) вычисляет функцию $f : B^* \to B^*$ (где B – подмножество алфавита машины, не содержащее пустого символа), если для каждого w из области определения функции f результат работы M равен $f(w)$, а для каждого w не из области определения f машина M не останавливается на входе w.

Функция f называется *вычислимой машинами Тьюринга*, если есть такая МТ, которая вычисляет f.

Дискретная математика. Коллоквиум весна 2017. Теоремы

Орлов Никита, Тимофей Гутор, Данила Усачёв, Иван Петровский, Андрей Ткачев

12 марта 2017 г.

Содержание

Теорема 1

Теорема. *Пусть* $(\Omega,\mathfrak{F},\mathcal{P})(\Omega,\mathfrak{F},\mathcal{P})$ *— вероятностное пространство. Тогда для произвольных событий* $A_1,A_2,\ldots,A_nA_1,A_2,\ldots,A_n$ *справедлива формула*

$$\mathcal{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_i \mathcal{P}(A_i) - \sum_{i<j} \mathcal{P}(A_i \cap A_j) + \sum_{i<j<k} \mathcal{P}(A_i \cap A_j \cap A_k) + \ldots + (-1)^{n-1}\,\mathcal{P}\left(\bigcap_{i=1}^{n} A_i\right).\mathcal{P}\left(\bigcup_{i=1}^{n}\right.$$

$\sum_i \mathcal{P}(A_i) - \sum_{i<j} \mathcal{P}(A_i \cap A_j) + \sum_{i<j<k} \mathcal{P}(A_i \cap A_j \cap A_k) + \ldots + (-1)^{n-1}\,\mathcal{P}\left(\bigcap_{i=1}^{n} A_i\right)$.

Доказательство. Её можно получить из принципа включений-исключений в форме индикаторных функций:

$$\mathbf{1}_{\bigcup_i A_i} = \sum_i \mathbf{1}_{A_i} - \sum_{i<j} \mathbf{1}_{A_i \cap A_j} + \sum_{i<j<k} \mathbf{1}_{A_i \cap A_j \cap A_k} + \ldots + (-1)^{n-1}\,\mathbf{1}_{A_1 \cap \ldots \cap A_n}.\mathbf{1}_{\bigcup_i A_i} = \sum_i \mathbf{1}_{A_i} - \sum_{i<j} \mathbf{1}_{A_i \cap A_j} +$$

$\sum_{i<j<k} \mathbf{1}_{A_i \cap A_j \cap A_k} + \ldots + (-1)^{n-1}\,\mathbf{1}_{A_1 \cap \ldots \cap A_n}$.

Пусть A_iA_i — события вероятностного пространства $(\Omega,\mathfrak{F},\mathcal{P})(\Omega,\mathfrak{F},\mathcal{P})$, то есть $A_i \in \mathfrak{F}A_i \in \mathfrak{F}$. Возьмем математическое ожидание $\mathcal{M}\mathcal{M}$ от обеих частей этого соотношения, и, воспользовавших линейностью математического ожидания и равенством $\mathcal{P}(A) = \mathcal{M}(\mathbf{1}_A)\mathcal{P}(A) = \mathcal{M}(\mathbf{1}_A)$ для произвольного события $A \in \mathfrak{F}A \in \mathfrak{F}$, получим формулу включения-исключения для вероятностей.

[:|||:]

Теорема 2

Теорема. *Условную вероятность* $Pr[A|B]$ *можно вычислить по формуле Байеса:*

$$Pr[A|B] = \frac{Pr[B|A]}{Pr[B]} \cdot Pr[A]$$

Доказательство.

$$Pr[A|B] = \frac{Pr[B|A]}{Pr[B]} \cdot Pr[A]$$

$$\Updownarrow$$

$$Pr[A|B] \cdot Pr[B] = Pr[B|A] \cdot Pr[A]$$

$$\Updownarrow$$

$$\frac{Pr[A \cap B]}{Pr[B]} \cdot Pr[B] = \frac{Pr[B \cap A]}{Pr[A]} \cdot Pr[A]$$

$$\Updownarrow$$

$$Pr[A \cap B] = Pr[B \cap A]$$

Т.к. $A \cap B = B \cap A$, то последнее равенство верно, а значит верна формула Байеса. **[:|||:]**

Теорема 3

Теорема. *Условной вероятностью события A при условии события B называется* $\mathbb{P}(A \mid B) = \dfrac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$*, где* $\mathbb{P}(A \cap B)\mathbb{P}(A \cap B)$ *— вероятность наступления обоих событий сразу.*

Доказательство. Пусть ровно r исходов события B входят и в событие A. Исходы события B уже реализовались В данном испытании произошло одно из t событий, входящих в B. Все элементарные события равновероятны, следовательно, для данного испытания вероятность наступления произвольного элементарного события, входящего в B равна 1/t. Тогда по классическому определению вероятности, в данном испытании событие A произойдет с вероятностью r/t. $P(A|B) = \frac{\frac{r}{m}}{\frac{t}{m}} = \frac{P(AB)}{PB}$

[:|||:]

Теорема 4

Теорема. *Математическое ожидание E линейно.*

Доказательство. Пусть ξ и η — случайные величины, заданные на одном вероятностном пространстве. Тогда выполняется равенство

$$E(\xi + \eta) = \sum_w (\xi(w) + \eta(w))p(w) = \sum_w \xi(w)p(w) + \sum_w \eta(w)p(w) = E(\xi) + E(\eta)$$

То есть математическое ожидание суммы случайных величин равно сумме математического ожидания каждой из этих величин. Пусть теперь ξ — случайная величина, α — действительное число. Тогда выполняется равенство

$$E(\alpha \cdot \xi) = \sum_w (\alpha \cdot \xi(w)p(w)) = \alpha \cdot \sum_w \xi(w)p(w) = \alpha \cdot E(\xi)$$

То есть математическое ожидание произведения константы и случайной величины равно произведению этой константы и математического ожидания самой величины.
Таким образом, линейность математического ожидания доказана. **[:|||:]**

Теорема 5

Неравенство Маркова в теории вероятностей дает оценку вероятности, что случайная величина превзойдет по модулю фиксированную положительную константу, в терминах её математического ожидания. Получаемая оценка обычно груба, однако она позволяет получить определённое представление о распределении, когда последнее не известно явным образом.

Теорема. *Пусть случайная величина X:* $\Omega \to \mathbb{R}+$ *определена на вероятностном пространстве* $(\Omega, F, \mathbb{R})$*, и ее математическое ожидание* $\mathbb{E}|\xi| < \infty$*. Тогда* $\forall\ x > 0\ \ \mathbb{P}(|\xi| \geq x) \leq \frac{\mathbb{E}|\xi|}{x}$

Доказательство. Возьмем для доказательства следующее понятие:

Пусть A - некоторое событие. Назовем индикатором события A случайную величину I, равную единице если событие A произошло, и нулю в противном случае. По определению величина I(A) имеет распределение Бернулли с параметром

$p = \mathbb{P}(I(A) = 1) = \mathbb{P}(A)$,

и ее математическое ожидание равно вероятности успеха $p = \mathbb{P}(A)$. Индикаторы прямого и противоположного событий связаны равенством $I(A) + I(\overline{A}) = 1$. Поэтому $|\xi| = |\xi| * I(|\xi| < x) + |\xi| * I(|\xi| \geq x) \geq |\xi| * I(|\xi| \geq x) \geq x * I(|\xi| \geq x)$. Тогда $\mathbb{E}|\xi| \geq \mathbb{E}(x * I(|\xi| \geq x)) = x * \mathbb{P}(|\xi| \geq x)$.

Разделим обе части на x:

$\mathbb{P}(|\xi| \geq x) \leq \frac{\mathbb{E}|\xi|}{x}$

Пример:

Ученики в среднем опаздывают на 3 минуты. Какова вероятность того, что ученик опоздает на 15 минут и более? Дать грубую оценку сверху.

$\mathbb{P}(|\xi| \geq 15) \leq 3/15 = 0.2$

[:|||:]

Теорема 6

Теорема. *Подмножество счетного множества конечно либо счетно.*

Доказательство. Рассмотрим счетное можество A и его подмножество A'. Выпишем элементы в последовательность

$$a_0, a_1, a_2, \ldots$$

Вычеркнем элементы, которые не лежат в A'. В результате останется последовательность элементов A' – конечная либо бесконечная, а значит и подмножество будет конечным либо счетным. **[:|||:]**

Теорема. *Любое бесконечное множество содержит счетное подмножество.*

Доказательство. Построим такое подмножество. Первый элемент выберем произвольно. Затем, так как исходное множество все еще бесконечно, выберем второй элемент. На i шаге нужно выбрать из бесконечного множества $S \setminus \{a_0, \ldots, a_{i-1}\}$ элемент, что сделать возможно. В итоге получим бесконечную последовательность элементов, являющихся элементами счетного подмножества. **[:|||:]**

Теорема 7

Теорема. *Объединение счетного числа счетных или конечных множеств счетно или конечно*

Доказательство. Пусть имеется счётное число счётных множеств A1, A2, . . .

Расположив элементы каждого из них слева направо в последовательность (Ai = ai0, ai1, . . .) и поместив эти последовательности друг под другом, получим таблицу

a00 a01 a02 a03 . . .

a10 a11 a12 a13 . . .

a20 a21 a22 a23 . . .

a30 a31 a32 a33 . . .

. . . . . .

Теперь эту таблицу можно развернуть в последовательность, например, проходя по очереди диагонали: a00, a01, a10, a02, a11, a20, a03, a12, a21, a30, . . . Если множества Ai не пересекались, то мы получили искомое представление для их объединения.

Если пересекались, то из построенной последовательности надо выбросить повторения. Если множеств конечное число или какие-то из множеств конечны, то в этой конструкции части членов не будет — и останется либо конечное, либо счётное множество.

[:|||:]

Теорема 8

Теорема. *Декартово произведение счетных множеств счетно.*

Доказательство. Б.о.о. можно считать, что необходимо доказать счетность $\mathbb{N} \times \mathbb{N}$. Разобьем наше декартово произведение в объединение множеств вида $\{a_0\} \times \mathbb{N}$. Каждое такое множество счетно. В итоге декартово произведение разложилось в счетное объединение счетных множеств, а значит и само счетно. **[:|||:]**

Теорема 9

Теорема. *Множество бесконечных последовательностей нулей и единиц несчетно.*

Доказательство. Предположим, что оно счетно, значит его можно пронумеровать. Тогда построим таблицу последовательностей.

$$\begin{matrix} a_{00} & a_{01} & a_{02} & \dots \\ a_{10} & a_{11} & a_{12}\dots & \\ a_{20}a_{21}a_{22}\dots & & & \end{matrix}$$

Теперь рассмотрим диагональную последовательность $a_{00}a_{11}a_{22}...$ и заменим в ней все биты на противоположные. Такая последовательность отличается от любой a_i в i-й позиции, значит этой последовательности нет в списке, получили противоречие. Значит это множество несчетно.

Теперь докажем, что множество бесконечных последовательностей нулей и единиц равномощно отрезку [0;1], то есть имеет мощность континуум.

Из курса анализа известно, что каждое число из [0;1] можно представить в виде бесконечной двоичной дроби. Делается это так: первый бит после запятой равен 0, если x лежит в левой половине отрезка [0,1] и равен 1, если в правой. И так далее. Делим отрезок пополам и смотрим, куда попал x. Но это не совсем биекция. Такие последовательности как 0,1001111... и 0, 101000... соотвествуют одному и тому же числу. Чтобы исправить это, надо исключить последовательности, в которых начиная с некоторого момента все цифры равны 1 (кроме 0.111111...). Но таких последовательностей счетное множество, так что их добавление не меняет мощность множества. **[:|||:]**

Теорема 10

Если для множества A и B существует инъекция из A в B и инъекция из B в A, то существует и биекция между A и B. Доказательство. Пусть $f : A \to B$ и $g : B \to A$ инъекции. Рассмотрим орграф с вершинами $A \cup B$. Для точек $x \in A$ и $y \in B$ проводим ребро из x в y, если $f(x) = y$ и ребро из y в x, если $g(y) = x$. По построению из каждой точки выходит ровно одно ребро. А так как функции инъективны, то и входит не больше одного.
Разобьем граф на компоненты связности, забыв об ориентации ребер, и рассмотрим каждую компоненту отдельно. Для каждой компоненты есть три варианта: Компонента может быть циклом из стрелок, бесконечной цепочкой стрелок, начинающейся в некоторой вершине или бесконечной в обе стороны цепочкой стрелок.
В нашем графе вершины бывают "левые"(из A) и "правые"(из B). Они чередуются, поэтому цикл может быть только четной длины и содержит поровну вершин из A и из B. Они чередуются, поэтому цикл может быть только четной длины и содержит поровну вершин из A и из B. Любое из отображений f и g может быть использовано чтобы построить биекцию между A и B вершинами цикла. То же самое верно для бесконечной в обе стороны цепочки. Если же цепочка бесконечна только в одну сторону, то для построения биекции годится только одно из отображений. Скажем, если она начинается с a, то нам годится только функция f (при которой a соответствует f(a)). Но в любом случае, одна из функций f и g годится, так что внутри каждой связной компоненты у нас есть биекция, и остается их объединить для всех связных компонент.

Теорема 11

Полнота стандартного базиса. Любое высказывание может быть выражено как дизъюнкция таких высказываний, у котороых ровно в одной строке стоит 1, а в остальных стоят нули. Действительно, выберем все строки таблицы высказывания, в которых стоят единицы. Для каждой такой строки образуем высказывание, которое истинно только в данной строке, а в

остальных ложно, дизъюнкция всех этих высказываний и будет выражать искомое.
Теперь научимся выражать через диъюнкции, конъюнкции и отрицания высказывания того вида, который использован в предыдущей конструкции. Чтобы получить высказывание , которое истинно ровно для одного произвольного набора логических значений, сделаем следующее: Если значение какой-то переменной равно единице, то включим эту переменную в высказывание, а если нулю, то включим ее отрицание. Построенная конъюнкция принимает значение 1 лишь тогда, когда все ее члены равны 1. По построению это происходит ровно на одном наборе значений переменных.

Теорема 12

Существование и единственность полинома Жегалкина. Сначала докажем по индукции, что любое произвольное высказывание $f(x_1 \dots x_n)$ можно выразить формулой со связкам $\wedge, \oplus, 1$. База индукции $n = 1$. Константа 1 уже есть. 0 выражается как $1 \oplus 1$.
Пусть утверждение доказано для всех составных высказываний от n элементарных высказываний. Докажем выразимость для составных высказываний от $n+1$ элементарного высказывания. Для этого по высказыванию $f(x_1, \dots, x_n+1)$ определим два высказывания от n элементарных высказываний, а именно $f_0(x_1, \dots x_n) = f(x_1, \dots, x_n, 0)$ и $f_1(x_1, \dots, x_n) = f(x_1, \dots, x_n, 1)$. По предположению индукции f_0 и f_1 выражаются через базис Жегалкина. Выразим теперь f (разложение Рида): $f = ((1 \oplus x_{n+1}) \wedge f_0) \oplus (x_{n+1} \wedge f_1)$. Действительно, при $x_{n+1} = 0$ обращается в 0 второе слагаемое, при $x_{n+1} = 1$ - первое. В любом случае получаем совпадение левой и правой частей равенства.
Теперь докажем единственность. Заметим, что различных булевых функций от n переменных 2^{2^n} штук. При этом конъюнкций вида $x_{i_1} \dots x_{i_k}$ существует ровно 2^n, так как из n возможных сомножителей каждый или входит в конъюнкцию, или нет. В полиноме у каждой такой конъюнкции стоит 0 или 1, то есть существует 2^{2^n} различных полиномов Жегалкина от n переменных.
Теперь достаточно лишь доказать, что различные полиномы реализуют различные функции. Предположим противное. Тогда приравняв два различных полинома и перенеся один из них в другую часть равенства, получим полином, тождественно равный нулю и имеющий ненулевые коэффициенты. Тогда рассмотрим слагаемое с единичным коэффициентом наименьшей длины, то есть с наименьшим числом переменных, входящих в него (любой один, если таких несколько). Подставив единицы на места этих переменных, и нули на места остальных, получим, что на этом наборе только одно это слагаемое принимает единичное значение, то есть нулевая функция на одном из наборов принимает значение 1. Противоречие. Значит, каждая булева функция реализуется полиномом Жегалкина единственным образом.

Теорема 13

Теорема. *Существует булева фукнция от n переменных схемной сложности* $\Omega(\frac{2^n}{n})$

Доказательство. Докажем, что всякую фукнцию можно вычислить схемой размера не больше $O(n2^n)$.

Для всякого $a \in \{0,1\}^n$ введем фукнцию $f_a(x)$, такую что

$$f_a(x) = \begin{cases} 1, & x = a \\ 0, & \text{иначе} \end{cases}$$

Введем обозначение $x^1 = x, x^0 = \overline{x}$. Тогда такая функция может быть записана следующим образом:

$$f_a(x) = \bigwedge i = 1^n x_i^{a_i}$$

Тогда для произвольной f:

$$f(x) = \bigvee_{a \in f^{-1}(1)} f_a(x)$$

Сначала наша схема должна вычислить отрицание всех элементов, потом вычислить все функции f_a. Для вычисления каждой потребуется $n-1$ раз применить конъюнкцию. Всего в итоге получится $2^n(n-1)$ элемент. В итоге нужно будет взять дизъюнкцию нужных функций. Получим 2^n элементов, и суммарно не более $O(n2^n)$.

[:|||:]

Теорема 14

Теорема. *Верхняя оценка $O(n2^n)$ схемной сложности булевой функции от n переменных.*

Доказательство. Для всякого $a \in 0,1^n$ рассмотрим функцию $f_a : 0,1^n \to 0,1$, такую что $f_a(x) = 1$ тогда и только тогда, когда $x = a$. Будет удобно ввести обозначение $x^1 = x$ и $x_0 = \neg x$. Тогда функцию f_a можно записать формулой

$$f_a(x) = \bigwedge_{i=1}^{n} x_i^{a_i},$$

где $x = (x_1, \ldots, x_n)$ и $a = (a_1, \ldots, a_n)$. Для произвольной функции f уже не сложно записать формулу через функции

$$f(x) = \bigvee_{a \in f^{-1}(1)} f_a(x).$$

Теперь эти формулы можно переделать в схему. Наша схема сначала будет вычислять отрицания всех переменных, на это нужно n элементов. После этого можно вычислить все функции f_a. Для вычисления каждого нужно $n-1$ раз применить конъюнкцию. Всего получается $2^n(n-1)$ элемент. Наконец, для вычисления f нужно взять дизъюнкцию нужных функций f_a, на это уйдет не более 2^n элементов (всего различных функций f_a ровно 2^n – над каждым аргументом отрицание либо есть, либо нет). Суммарно в нашей схеме получается $O(n2^n)$ элементов. **[:|||:]**

Теорема 15

Схема умножения n-битовых чисел за $O(n^2)$.

Пусть на вход подаются два числа $x = x_{n-1} \dots x_1 x_0$ и $y = y_{n-1} \dots y_1 y_0$. Мы хотим вычислить $z = x \cdot y$. Заметим, что z имеет не больше $2n$ разрядов. Действительно, $x, y < 2^n$, так что $z = x \cdot y < 2^{2n}$, а значит для его записи достаточно $2n$ разрядов.

Для вычисления z воспользуемся школьным методом. В нем умножение двух чисел сводится к сложению n чисел. Действительно, чтобы умножить x на y достаточно для всякого $i = 0, \dots, n-1$ умножить x на y_i, приписать в конце числа i нулей и затем сложить все полученные числа. Умножение x на y_i легко реализуется с помощью n конъюнкций. После этого остается сложить n чисел длины не более $2n$. Для этого мы можем $n-1$ раз применить схему для сложения. Размер каждой схемы для сложений линейный, так что суммарная сложность схемы для умножения получается $O(n^2)$.

Теорема 16

Схема проверки связности графа на n вершинах полиномиального размера.

Пусть матрица A - матрица смежности графа с единицами на главной диагонали. Можно показать, что на пересечении строки i и столбца j матрицы A^k записано число путей длины k из вершины v_i в вершину v_j. Теперь рассмотрим матрицу A', которая отличается от матрицы A тем, что у нее стоят единицы на главной диагонали.

Заметим следующий факт: если между двумя вершинами есть путь длины меньше n - 1, то есть и путь длины ровно $n-1$, достаточно добавить нужное количество петель. То есть надо рассмотреть матрицу $(A')^{n-1}$. Если в яйчеках нет нулей - граф связен, иначе нет. Теперь опишем схему.

На вход схема получает матрицу смежности A'. Схема последовательно вычисляет булевы степени этой матрицы $(A')^2, \dots, (A')^{n-1}$. Затем схема вычисляет конъюнкцию всех ячеек матрицы $(A')^{n-1}$ и подает ее на выход.

Оценим размер схемы. Для булева умножения достаточно $n^2 \cdot O(n) = O(n^3)$ операций. Всего нам нужо $(n-1)$ умножений, так что для вычисления матрицы $(A')^{n-1}$ достаточно $O(n^4)$ операций. Для последнего этапа - конъюнкции нужно $O(n^2)$ операций. Итого получается $O(n^4) + O(n^2) = O(n^4)$ операций.

Теорема 17

Теорема. *Разрешимые множества перечислимы.*

Доказательство. Алгоритм перечисления множества A использует алгоритм разрешения множества A. Он перебирает все числа, начиная с 0; для каждого числа n вычисляет индикаторную функцию $\chi_A(n)$ и печатает число n, если полученное значение равно 1. Корректность такого алгоритма очевидна из определений. **[:|||:]**

Теорема 18

Теорема. *Множество M и его дополнение $\overline{M}$ разрешимы тогда и только тогда, когда M и $\overline{M}$ перечислимы.*

Доказательство.

Необходимость:
Пусть M и $\overline{M}$ разрешимы. Случаи, когда $M = \mathbb{N}$ или $M = \varnothing$, тривиальны. Будем считать, что $M \neq \varnothing$ и M $\neq \mathbb{N}$. Тогда существуют такие a и b, что $a \in M$ и $b \in \overline{M}$. Поскольку M разрешимо, его характеристическая функция χ_M вычислима. Рассмотрим функцию

$$f(x) = \begin{cases} x \text{ при } \chi_M(x) = 1 \\ a \text{ при } \chi_M(x) = 0 \end{cases}$$

M является множеством значений f: ничего, кроме значений M, в $E(f)$, очевидно, быть не может, а для любого $m \in M$ верно, что $f(m) = m$. Аналогично, рассмотрим функцию

$$g(x) = \begin{cases} x \text{ при } \chi_M(x) = 0 \\ b \text{ при } \chi_M(x) = 1 \end{cases}$$

$\overline{M}$ является областью значений g. Таким образом, M и $\overline{M}$ перечислимы (перечисляющие алгоритмы могут быть, например, устроены так: последовательно для всех натуральных n, начиная с нуля, алгоритм выводит значение $f(n)$ или $g(n)$ соответственно).

Достаточность:
Пусть M и $\overline{M}$ перечислимы. Тогда существуют алгоритмы соответственно $\mathfrak{A}$ и $\mathfrak{B}$, с помощью которых могут быть получены все элементы этих множеств. Рассмотрим алгоритм, запускающий $\mathfrak{A}$ и $\mathfrak{B}$ параллельно, который выводит сначала первое число, полученное $\mathfrak{A}$, затем — первое число, полученное $\mathfrak{B}$, затем — второе число, полученное $\mathfrak{A}$, и так далее. Такой алгоритм будет являться перечисляющим алгоритмом $\mathbb{N}$, который получает элементы M на нечётных выводах и элементы $\overline{M}$ — на чётных. Соответсвенно, для любого элемента x верно, что он будет выведен рассматриваемым алгоритмом за конечное число шагов. Если он был выведен как нечётный по счёту вывод, то $\chi_M(x) = 1$, если как чётный — $\chi_M(x) = 0$. Таким образом, χ_M вычислима, а значит, M и $\overline{M}$ разрешимы. **[:|||:]**

Теорема 19

Теорема. *Перечислимые множества являются множествами значений вычислимых функций.*

Доказательство. Пусть M — перечислимое множество. Тогда существует алгоритм $\mathfrak{A}$, выводящий все его элементы. Рассмотрим алгоритм, который принимает на вход натуральное число n, после чего запускает $\mathfrak{A}$ и считает его выводы. Дойдя до n-го по счёту (начиная с 0) вывода, алгоритм останавливается, возвращая n-й вывод алгоритма $\mathfrak{A}$ как результат своей работы.
Множество значений функции, которую вычисляет вышеописанный алгоритм, будет совпадать с множеством чисел, выводимых $\mathfrak{A}$, то есть с M.

[:|||:]

Теорема 20

Теорема. *Перечислимые множества являются множествами значений всюду определённых вычислимых функций.*

Доказательство. Пусть M — перечислимое множество. Тогда существует алгоритм $\mathfrak{A}$, выводящий все его элементы. Рассмотрим алгоритм, который принимает на вход натуральное число n, после чего запускает $\mathfrak{A}$ и считает его выводы. Дойдя до n-го по счёту (начиная с 0) вывода, алгоритм останавливается, возвращая n-й вывод алгоритма $\mathfrak{A}$ как результат своей работы.
Множество значений функции f, которую вычисляет вышеописанный алгоритм, будет совпадать с множеством чисел, выводимых $\mathfrak{A}$, то есть с M. Если множество M бесконечно, то f также будет всюду определённой по построению. Если же M конечно, рассмотрим функцию $f_1(x) = f(x \bmod (l+1))$, где l — номер вывода $\mathfrak{A}$, после которого количество различных выведенных $\mathfrak{A}$ элементов станет равно $[M]$. Значение l будет конечным, так как любой элемент M выводится $\mathfrak{A}$ за конечное число шагов. Данная функция будет всюду определённой, поскольку $\mathfrak{A}$ до своей остановки совершает не менее l шагов, и множество её значений будет совпадать с M, поскольку по построению в множестве её значений $[M]$ различных элементов, и все они являются результатом работы $\mathfrak{A}$.

[:|||:]

Теорема 21

Теорема. *Множества значений всюду определённых функций перечислимы.*

Доказательство. Пусть $M = f(\mathbb{N})$ — множество значений некоторой всюду определённой функции f. Рассмотрим алгоритм, последовательно выводящий для каждого натурального числа n, начиная с 0, значение $f(n)$. Он будет являться перечисляющий алгоритмом для M: для любого $m \in M$ верно, что $\exists x \in \mathbb{N} : f(x) = m$, следовательно, вышеописанный алгоритм выведет m на своём x-ом шаге. **[:|||:]**

Теорема 22

Теорема. *Множество значений всюду опрелённой вычислимой функции является областью определения вычислимой функции.*

Доказательство. Пусть f — всюду определённая вычислимая функция. Рассмотрим алгоритм, принимающий на вход натуральное число x, который последовательно вычисляет значения $f(n)$ для всех натуральных n, начиная с 0, и, если полученное в какой-то момент значение равно x, выводит 1. Если $x \in E(f)$, то $\exists m \in \mathbb{N} : f(m) = x$. Тогда вышеописанный алгоритм остановится за конечное число шагов: он завершит свою работу, вычислив значения $f(n)$ для всех $n \leqslant m$, а для этого требуется конечное число шагов, поскольку f вычислима и всюду определена. Если же $x \notin E(f)$, то данный алгоритм никогда не остановится, поскольку условие его остановки — существование такого $m \in \mathbb{N}$, что $f(m) = x$. Таким образом, функция, вычисляемая вышеописанным алгоритмом, определена в точности на $E(f)$. **[:|||:]**

Теорема 23

Теорема. *Область определения вычислимой функции является множеством значений вычислимой функции.*

Доказательство. Пусть S – область определения некоторой вычислимой функции f, а p – номер программы, вычисляющей f в нумерации U. Рассмотрим функцию g:

$$g(x,t) = \begin{cases} x & F(p,x,t) = 1 \\ - & F(p,x,t) = 0 \end{cases}$$

Если $x \in S$, то $x = g(x,t)$ для некоторого t. И обратно, если $x = g(x,t)$ для некоторого t, то $U(p,x)$ определена, а значит, определена и $f(x)$.

Мы представили S как множество значений функции от двух натуральных аргументов. Чтобы перейти к функциям одного аргумента, используем вычислимую биекцию $c : N \times N \to N$ и выразим S как $S = g \circ c^{-1}(N)$. **[:|||:]**

Теорема 24

Теорема. *Непустое множество значений вычислимой фукнции является множеством значений всюду определенной вычислимой фукнции.*

Доказательство. Пусть $S = f(\mathbb{N})$ для некоторой вычислимой f. Пусть $f(x) = U(p, x)$ для некоторой у.в.ф. U, для которой существует отладочная функция F.

Пусть g – всюду определенная фукнция $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$, определенная следующим образом:

$$g(x,t) = \begin{cases} U(p,x), & F(p,x,t) = 1 \\ a, & \text{иначе} \end{cases}$$

Множество значений g совпадает с S: если $y = g(x,t)$, то $y = a \in S$ или $y = U(p,x) = f(x) \in S$. В другую сторону: пусть $y = f(x) = U(p,x)$. На паре (p,x) фукнция определена, значит существует t, такое что $F(p,x,t) = 1 \Rightarrow y = g(x,t)$. Получили, что множество S представимо в виде множества занчений тотальной функции от двух аргументов. Осталось перейти к функции от одного аргумента, использовав любую вычислимую биекцию.

[:|||:]

Теорема 25

Теорема. *Множество S, являющееся областью определения универсальной фукнции является перечислимым, но неразрешимым множеством.*

Доказательство. Перечислимость. Пусть S – область определения некоторой вычислимой фукнции f. Такая область перечислима. Обозначим через p номер функции в нумерации U. Получим что

$$S = \{x : U(p,x) \text{определена}\}$$

Неразрешимость. Если бы оно было разрешимо, что из алгоритма разрешения получался бы алгоритм разрешения любого перечислимого множества. **[:|||:]**

Теорема 26

TO BE WRITTEN. DEADLINE: 12.03.2017

Теорема 27

Теорема. *Функция вычислима тогда и только тогда, когда ее график перечислим.*

Доказательство. 1) Пусть функция f вычислима. Тогда возможно перечислить ее график через функцию отладки: будем перечислять $\mathbb{N}^3$ и 2) Пусть график функции f перечислим. Тогда алгоритм ее вычисления тривиален: перечисляем график и на каждой выданной паре будем сравнивать вход с первой координатой. Если функция определена, значит она когда-нибудь **[:|||:]**

Теорема 29

Определения:

- **Свойством** называется некоторое подмножество множества F всевозможных вычислимых функций.
- Свойство A называется **нетривиальным**, если $A \neq F$ и $A \neq \varnothing$.

Пусть U — главная универсальная функция.

Теорема. *Теорема Успенского-Райса: для любого нетривиального свойства A множество $\{n \,|\, U(n,x) \in A\}$ неразрешимо.*

Доказательство. Пусть A — нетривиальное свойство, α — нигде не определённая функция. Без ограничения общности предположим, что $\alpha \in A$ (если это не так, рассмотрим $\overline{A}$: A разрешимо тогда и только тогда, когда $\overline{A}$ разрешимо). Пусть $\beta \in \overline{A}$ — некоторая вычислимая функция (такая функция существует, так как A нетривиально). Рассмотрим произвольное перечислимое, но не разрешимое множество K и функцию $V(n,x)$, заданную следующим образом:

$$V(n,x) = \begin{cases} \beta(x) \text{ при } n \in K \\ \alpha(x) \text{ при } n \notin K \end{cases}$$

Данная функция вычисляется алгоритмом, который запускает перечисляющий алгоритм $\mathfrak{A}$ множества K, каждый вывод $\mathfrak{A}$ сравнивает с n и в случае равенства останавливается и возвращает $\beta(x)$ как результат своей работы. При $n \in K$ алгоритм $\mathfrak{A}$ выведет n через конечное число шагов, и вышеописанный алгоритм остановится за конечное число шагов, а при $n \notin K$ вышеописанный алгоритм никогда не остановится, так как условие есго остановки (равенство некотрого вывода $\mathfrak{A}$ и n) никогда не будет выполнено.
Поскольку U — главная универсальная функция, существует всюду определённая вычислимая функция s такая, что $V(n,x) = U(s(n),x)$ для любых x и n. Предположим, что $\{n \,|\, U(n,x) \in A\}$ — разрешимое множество. Заметим, что $n \in K \Leftrightarrow U(s(n),x) \notin A$ по определению $V(n,x)$. Если $\{n \,|\, U(n,x) \in A\}$ разрешимо, то разрешимо и $\{n \,|\, U(s(n),x) \in A\}$ (поскольку s — всюду определённая вычислимая функция), а следовательно, и K. Однако K — неразрешимое множество. Значит, предположение неверно, и $\{n \,|\, U(n,x) \in A\}$ неразрешимо для любого нетривиального A. **[:|||:]**

Теорема 30

Пусть U — главная универсальная функция, p — всюду определённая вычислимая функция.

Теорема. *Теорема о неподвижной точке: существует такое t, что $U(t,x) = U(p(t),x)$ при любых x.*

Доказательство. Рассмотрим функцию $a(x) = U(x, x)$. Поскольку U — главная универсальная функция, $U(a(x), y) = V(x, y) = U(s(x), y)$ для любого y, где s — некоторая всюду определённая вычислимая функция. Так как композиция $p \circ s$ — всюду определённая вычислимая функция, существует $C(p, s) = q$ такое, что $U(q, x) = U(p, U(s, x))$ (в данном случае под p и s понимаются U-номера соответствующих функций, то есть $U(p, U(s, x))$ — это то же самое, что и $p(s(x))$).
Докажем, что $s(q)$ является непожвижной точкой для функции p: заметим, что $U(p(s(q)), x) = U(U(q, q), x)$ по определению q. В свою очередь, $U(U(q, q), x) = U(a(q), x) = U(s(q), x)$ по определению $a(x)$. Таким образом, $U(p(s(q)), x) = U(s(q), x)$, что нам и требовалось. **[:|||:]**

=======

Теорема 29

Определения:

- **Свойством** называется некоторое подмножество множества F всевозможных вычислимых функций.
- Свойство A называется **нетривиальным**, если $A \neq F$ и $A \neq \varnothing$.

Пусть U — главная универсальная функция.

Теорема. *Теорема Успенского-Райса: для любого нетривиального свойства A множество $\{n \,|\, U(n,x) \in A\}$ неразрешимо.*

Доказательство. Пусть A — нетривиальное свойство, α — нигде не определённая функция. Без ограничения общности предположим, что $\alpha \in A$ (если это не так, рассмотрим $\overline{A}$: A разрешимо тогда и только тогда, когда $\overline{A}$ разрешимо). Пусть $\beta \in \overline{A}$ — некоторая вычислимая функция (такая функция существует, так как A нетривиально). Рассмотрим произвольное перечислимое, но не разрешимое множество K и функцию $V(n, x)$, заданную следующим образом:

$$V(n, x) = \begin{cases} \beta(x) \text{ при } n \in K \\ \alpha(x) \text{ при } n \notin K \end{cases}$$

Данная функция вычисляется алгоритмом, который запускает перечисляющий алгоритм $\mathfrak{A}$ множества K, каждый вывод $\mathfrak{A}$ сравнивает с n и в случае равенства останавливается и возвращает $\beta(x)$ как результат своей работы. При $n \in K$ алгоритм $\mathfrak{A}$ выведет n через конечное число шагов, и вышеописанный алгоритм остановится за конечное число шагов, а при $n \notin K$ вышеописанный алгоритм никогда не остановится, так как условие есго остановки (равенство некотрого вывода $\mathfrak{A}$ и n) никогда не будет выполнено.
Поскольку U — главная универсальная функция, существует всюду определённая вычислимая функция s такая, что $V(n,x) = U(s(n), x)$ для любых x и n. Предположим, что $\{n \,|\, U(n, x) \in A\}$ — разрешимое множество. Заметим, что $n \in K \Leftrightarrow U(s(n), x) \notin A$ по определению $V(n, x)$. Если $\{n \,|\, U(n, x) \in A\}$ разрешимо, то разрешимо и $\{n \,|\, U(s(n), x) \in A\}$ (поскольку s — всюду определённая вычислимая функция), а следовательно, и K. Однако K — неразрешимое множество. Значит, предположение неверно, и $\{n \,|\, U(n, x) \in A\}$ неразрешимо для любого нетривиального A. **[:|||:]**

Теорема 30

Пусть U — главная универсальная функция, p — всюду определённая вычислимая функция.

Теорема. *Теорема о неподвижной точке: существует такое t, что $U(t,x) = U(p(t),x)$ при любых x.*

Доказательство. Рассмотрим функцию $a(x) = U(x,x)$. Поскольку U — главная универсальная функция, $U(a(x),y) = V(x,y) = U(s(x),y)$ для любого y, где s — некоторая всюду определённая вычислимая функция. Так как композиция $p \circ s$ — всюду определённая вычислимая функция, существует $C(p,s) = q$ такое, что $U(q,x) = U(p,U(s,x))$ (в данном случае под p и s понимаются U-номера соответствующих функций, то есть $U(p,U(s,x))$ — это то же самое, что и $p(s(x))$).
Докажем, что $s(q)$ является непожвижной точкой для функции p: заметим, что $U(p(s(q)),x) = U(U(q,q),x)$ по определению q. В свою очередь, $U(U(q,q),x) = U(a(q),x) = U(s(q),x)$ по определению $a(x)$. Таким образом, $U(p(s(q)),x) = U(s(q),x)$, что нам и требовалось. **[:|||:]**