Конспект по теории вычислимости
IV семестр, 2021 год
Современное программирование, факультет математики и компьютерных наук, СПбГУ
(лекции Пузыниной Светланы Александровны)

Тамарин Вячеслав

June 10, 2021

Contents

Исходный код на https://github.com/tamarinvs19/theory_university

Некоторые доказательства были опущены на лекции, но написаны мной. Они выделены оранжевыми символами:

□ Исправляйте, дополняйте. Чем меньше недоказанных утверждений, тем лучше! ■

Index

Chapter 1

Вычислимость. Система вычислимости по Клини

1.1 Рекурсивные функции

Лекция 1
11 feb

Определение 1

Пусть функция $f\colon \mathbb{N}^k \to \mathbb{N}$, $k \in \mathbb{N}$, где $\mathbb{N} = \{0, 1, 2, \ldots\}$[a]. Такая функция называется k-***местной частичной функцией***. Если $k = 0$, то $f = \text{const}$.

[a]Мы здесь считаем ноль натуральным числом

1.1.1 Простейшие функции

Простейшими будем называть следующие функции:

- Нуль местный нуль — функция без аргументов, возвращающая 0;
- Одноместный нуль — $0(x) = 0$;
- Функция следования — $s(x) = x + 1$;
- Функция выбора (проекция) — $I_n^m(x_1, \ldots x_n) = x_m$

1.1.2 Операторы

Определим три оператора:

Определение 2

- Функция f ***получается оператором суперпозиции*** из функций h и g_i, где

$$h(y_1, \ldots, y_m),\ g_i(x_1, \ldots, x_n);\ 1 \leqslant i \leqslant m,$$

  если

$$f = h(g_1(x_1, \ldots, x_n), \ldots g_m(x_1, \ldots, x_n)).$$

  Оператор обозначается **S**.

- Функция $f^{(n+1)}$[a] ***получается оператором примитивной рекурсией*** из $g^{(n)}$ и $h^{(n+2)}$, если

$$\begin{cases} f(x_1, \ldots x_n, 0) = g(x_1, \ldots x_n) \\ f(x_1, \ldots x_n, y+1) = h(x_1, \ldots x_n, y, f(x_1, \ldots x_n, y)) \end{cases}$$

  Оператор обозначается **R**.

- Функция f задается ***оператором минимизации*** (**M**), если она получается из функции g:

$$f(x_1, \ldots x_n) = \mu y\big[g(x_1, \ldots x_n, y) = 0\big] =$$
$$= \begin{cases} y & g(x_1, \ldots x_n, y) = 0 \land g(x_1, \ldots x_n, i)^{b} \neq 0\ \forall i < y \\ \uparrow^{c} & else \end{cases}$$

[a]Здесь и далее $f^{(n)}$ обозначается функция, принимающая n аргументов, то есть n-местная
[b]подразумевается, что функция определена в этих точках
[c]не определена

Пример 1.1.1

$$x - y = \begin{cases} x - y, & x \geqslant y \\ \uparrow, & x < y \end{cases}$$

Можно задать, используя оператор минимизации:

$$x - y = \mu z[|(y+z) - x| = 0].$$

1.1.3 Функции

Определение 3: Примитивно рекурсивная функция

Функция f называется ***примитивно рекурсивной*** (**ПРФ**), если существует последовательность таких функций $f_1, \ldots f_k$, что все f_i либо простейшие, либо получены из предыдущих $f_1, \ldots f_{i-1}$ с помощью одного из операторов **S** и **R** и $f = f_k$.

Пример 1.1.2

Докажем, что $f(x,y) = x + y$ — **ПРФ**. По **R** можем получить f так:

$$\begin{cases} f(x,0) & = x = I_1^1(x) \\ f(x,y+1) & = (x+y) + 1 = s(f(x,y)) = s(I_3^3(x,y,f(x,y)) \end{cases}$$

Теперь построим последовательность функций f_i, где последним элементом будет f, полученный с помощью **R**:

$$g = I_1^1,\ s,\ I_3^3,\ h = \mathbf{S}(s, I_3^3),\ f = \mathbf{R}(g,h).$$

Определение 4: Частично рекурсивная функция

Функция f называется ***частично рекурсивной функцией*** (**ЧРФ**), если существует последовательность функций $f_1, \ldots f_k$, таких что f_i либо простейшая, либо получается из предыдущих с помощью одного из операторов **S**, **R**, **M**.

Замечание. Частично рекурсивная функция может быть не везде определена. Примитивно рекурсивная определена везде.

Замечание. Существуют частично рекурсивные функции, которые всюду определены, но при этом не являются **ПРФ**.

Определение 5
Общерекурсивная функция — всюду определенная частично рекурсивная.

Пример 1.1.3
$\mu y[x + y + 1 = 0]$ — нигде не определена, но получается из последовательности других функций с помощью операторов.

Лемма 1. Следующие функции являются **ПРФ**:

1. $\mathrm{const}^{(n)}$
2. $x + y$
3. $x \cdot y$
4. x^y, где 0^0 можем определить, как хотим
5. $\mathrm{sg}(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0 \end{cases}$
6. $\overline{\mathrm{sg}}(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$
7. $x \dot{-} 1 = \begin{cases} u & x = 0 \\ x - 1 & x > 0 \end{cases}$
8. $x \dot{-} y = \begin{cases} 0 & x < y \\ x - y & else \end{cases}$
9. $|x - y|$

□

1. Сначала можем получить нужное число последовательной суперпозицией функции следования (получили константу от одной переменной), затем проецируем I_1^{n+1}, чтобы получить n переменных (первая - наша константа).
2. Доказали выше в примере 1.1.2.
3. $f(x, y) = xy$ определим так:
$$\begin{cases} f(x,0) & = 0 \\ f(x,y+1) & = f(x,y) + x \end{cases}$$
а складывать мы умеем.
4. $f(x, y) = x^y$:
$$\begin{cases} f(x,0) & = 1 = s(0) \\ f(x,y+1) & = f(x,y) * y \end{cases}$$
Умножать тоже можно по третьему пункту.
5. $\mathrm{sg}(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0 \end{cases}$
$$\begin{cases} \mathrm{sg}(0) & = 0 \\ \mathrm{sg}(x+1) & = 1 = s(0) \end{cases}$$

6. Аналогично
7. $f(x) = x \dot{-} 1$
$$\begin{cases} f(0) & = 0 \\ f(x+1) & = x = I_1^1(x) \end{cases}$$
8. $f(x,y) = x \dot{-} y$
$$\begin{cases} f(x,0) & = x = I_1^1(x) \\ f(x,y+1) & = f(x,y) \dot{-} 1 \end{cases}$$
9. $f(x,y) = |x-y| = (x \dot{-} y) + (y \dot{-} x)$

Замечание. Обычное вычитание не является **ПРФ**, так как не везде определено на $\mathbb{N}$.

1.1.4 Оператор ограниченной минимизации

Определение 6: Оператор ограниченной минимизации

Функция $f^{(n)}$ задается ***оператором ограниченной минимизации*** из функций $g^{(n+1)}$ и $h^{(n)}$, если
$$\mu y \leqslant h(\overline{x})[g(\overline{x},y) = 0]^{a}.$$
Это означает, что
$$f(\overline{x}) = \begin{cases} y & g(\overline{x},y) = 0 \land y \leqslant h(\overline{x}) \land g(\overline{x},i) \neq 0^{b}\ \forall i < y \\ h(\overline{x})+1 & else \end{cases}$$

[a] Здесь и далее $\overline{x} = x_1, \dots x_n$.
[b] Аналогично, подразумевается, что функция определена в этих точках

Утверждение. Пусть $g^{(n+1)}, h^{(n)}$ — примитивно рекурсивные функции, и $f^{(n)}$ получается из g и h с помощью ограниченной минимизации, то f тоже **ПРФ**.

Заметим, что f можно получить следующим образом:
$$f(\overline{x}) = \sum_{y=0}^{h(\overline{x})} \prod_{i=0}^{y} \mathrm{sg}(g(\overline{x},i)).$$
Внутреннее произведение равно единице только тогда, когда все $g(\overline{x},i) \neq 0$. Если для некоторого y обнуляется $g(\overline{x},y)$, то все произведения, начиная с $y+1$, будут равны нулю, поэтому просуммируем только y единиц. Если же такого y нет, получим сумму из $h(\overline{x})+1$ единицы. Именно это и нужно.

Проверим, что можно получить
$$a(\overline{x},y) = \sum_{i=0}^{y} g(\overline{x},i), \quad m(\overline{x},y) = \prod_{i=0}^{y} g(\overline{x},i)$$
с помощью примитивной рекурсии:
$$\begin{cases} a(\overline{x},0) & = g(\overline{x},0) \\ a(\overline{x},y+1) & = a(\overline{x},y) + g(\overline{x},y+1) \end{cases} \quad \begin{cases} m(\overline{x},0) & = g(\overline{x},0) \\ m(\overline{x},y+1) & = m(\overline{x},y) \cdot g(\overline{x},y+1) \end{cases}$$

Замечание. $0(x)$ можно исключить из определения простейших функций, так как ее можно получить с помощью оператора **R** для нульместного 0 и $I_2^2(x, y)$:

$$0(y) = \begin{cases} 0(0) & = 0 \\ 0(y+1) & = I_2^2(y, 0) \end{cases}$$

1.1.5 Предикаты

Определение 7

Предикат — условие задающее подмножество: $R \subset \mathbb{N}^k$.

Предикат называется ***примитивно рекурсивным (общерекурсивным)***, его характеристическая функция примитивно рекурсивная (общерекурсивная).

$$\chi_R(\overline{x}) = \begin{cases} 1, & \overline{x} \in R \\ 0, & \overline{x} \notin R \end{cases}$$

Утверждение.

- Если R, Q — примитивно рекурсивные (общерекурсивные) предикаты, то предикаты $P \vee Q$, $P \wedge Q$, $P \to Q$, $\neg P$ тоже примитивно рекурсивные (общерекурсивные).
- Предикаты $=, \leqslant, \geqslant, <, >$ тоже примитивно и общерекурсивны.

□

- Проверим, что характеристические функции примитивно / общерекурсивны:

$$\chi_{P \wedge Q}(\overline{x}) = \chi_P(\overline{x}) \cdot \chi_Q(\overline{x})$$
$$\chi_{P \vee Q}(\overline{x}) = \mathrm{sg}(\chi_P(\overline{x}) + \chi_Q(\overline{x}))$$
$$\chi_{P \to Q}(\overline{x}) = \overline{\mathrm{sg}}(\chi_P(\overline{x}) + \overline{\mathrm{sg}}(\chi_Q(\overline{x})))$$
$$\chi_{\neg P}(\overline{x}) = \overline{\mathrm{sg}}(\chi_P(\overline{x}))$$

- Аналогично выразим, через простейшие:

$$\chi_{x=y}(x) = \overline{\mathrm{sg}}(|x - y|) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$$
$$\chi_{x<y}(x) = \mathrm{sg}(x \dot{-} y)$$

  Остальные можем выразить также или через уже проверенные $<$ и $\neg$.

■

Лемма 2. Следующие функции являются примитивно рекурсивными:

1. $\left\lfloor \frac{x}{y} \right\rfloor$, считаем, что $\left\lfloor \frac{x}{0} \right\rfloor = x$
2. $\mathrm{Div}(x, y) = \begin{cases} 1, & y \mid x \\ 0, & else \end{cases}$
3. $\mathrm{Prime}(x) = \begin{cases} 1, & x \in \mathbb{P} \\ 0, & else \end{cases}$
4. $f(x) = p_x$, где p_x — x-тое простое число, $p_0 \coloneqq 2$
5. $\mathrm{ex}(i, x)$ — степень простого числа p_i разложении x, $\mathrm{ex}(i, 0) \coloneqq 0$

□

1. $f(x,y) = \left\lfloor \frac{x}{y} \right\rfloor$. Найдем минимальное k, что $f'(x,y,k) = yk > x$. Чтобы получить

$$f(x,y) = \min(k \mid f'(x,y,k)) - 1,$$

используем оператор минимизации:

$$f(x,y) = \mu k[\neg f'(x,y,k) = 0] - 1.$$

2. $\mathrm{Div}(x,y) = \left\lfloor \frac{x}{y} \right\rfloor \cdot y = x$

3. Определим $\mathrm{Div}'(x,y) = (y \leqslant 1) \vee (\neg \mathrm{Div}(x,y))$, эта функция проверяет, что число y не является нетривиальным делителем x.

   Теперь, используя ограниченную минимизацию, выразим $\mathrm{Prime}(x)$:

$$\mathrm{Prime}(x) = \left(\mu y \leqslant h(x)[\mathrm{Div}'(x,y) = 0]\right) = x, \text{ где } h(x) = x - 1.$$

   То есть мы посмотрели на все меньшие числа, если среди них найдется нетривиальный делитель, то число не простое.

4. Пусть $f'(x) =$ количество простых $\leqslant x$.

$$\begin{cases} f'(0) & = 0 \\ f'(x+1) & = \mathrm{Prime}(x+1) + f(x) \end{cases}$$

   Теперь можно вычислить $f(x)$: для этого определим функцию $g(x,y) = (f'(y) = x)$,

$$f(x) = \mu y[\neg g(x,y) = 0].$$

5. Чтобы найти степень вхождения простого числа p_i в x, сначала находим это простое число по номеру, затем находим минимальное k, что x не делится на p_i^k и вычитаем единицу.

■

1.1.6 Теоремы про рекурсии

Теорема 1.1.1 (Канторовская нумерация). Пусть $\pi\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$:

$$\pi(x,y) = \frac{1}{2}(x+y)(x+y+1) + y.$$

- Тогда для любого z существует единственное представление $z = \pi(x,y)$.
- Причем функции $x(z), y(z)$ примитивно рекурсивные.

□ Можно по-честному все посчитать и выразить $x(z),\ y(z)$. Пусть

$$\begin{aligned} w &= x + y \\ t &= \frac{1}{2}w(w+1) = \frac{w^2 + w}{2} \\ z &= t + y \end{aligned}$$

Решим квадратное уравнение, чтобы выразить w через t[1]:

$$w = \frac{-1 + \sqrt{8t+1}}{2}.$$

Запишем неравенство:

$$t \leqslant z = t + y < t + (w+1) = \frac{(w+1)^2 + (w+1)}{2}.$$

Аналогично выразим $w+1$ через z: имеем $z < \frac{(w+1)^2+(w+1)}{2}$, решаем неравенство, а далее вспоминаем, что все числа положительные и можно забыть про отрицательные корни. Отсюда

$$w = \frac{-1 + \sqrt{8t+1}}{2} \leqslant \frac{-1 + \sqrt{8z+1}}{2} < w + 1.$$

Тогда

$$\begin{aligned} w &= \left\lfloor \frac{-1 + \sqrt{8z+1}}{2} \right\rfloor \\ t &= \frac{w^2 + w}{2} \\ y &= z - t \\ x &= w - y \end{aligned}$$

Таким образом, мы выразили через z обе координаты. Единственный момент — нужно извлекать корень, в натуральную степень возводить мы умеем, поэтому можем с помощью ограниченной минимизации перебрать все меньшие числа, возвести их в квадрат и сравнить с нашим числом.

Теорема 1.1.2 (Возвратная рекурсия). Зафиксируем s. Пусть

$$\begin{cases} f(\overline{x}, 0) &= g(\overline{x}) \\ f(\overline{x}, y+1) &= h(\overline{x}, y, f(\overline{x}, t_1(y)), \ldots f(\overline{x}, t_s(y))) \end{cases}$$

где $\forall 1 \leqslant i \leqslant s\ t_i(y) \leqslant y, g^{(n)}, h^{(n+1+s)}, t_i^{(1)}$.
Тогда, если g, h, t_i — примитивно / общерекурсивные, то и f тоже.

Основная идея этой теоремы — можем использовать все ранее вычисленные значения функции, а не только предыдущее.

Построим с помощью примитивной рекурсии функцию $m(\overline{x}, y)$, которая возвращает закодированную последовательность $f(\overline{x}, i)$, $0 \leqslant i \leqslant y$.

Кодировать будем так: каждому $f(\overline{x}, i)$ будет соответствовать p_i (i-ое простое число) в степени $1 + f(\overline{x}, i)$.

Если мы построим эту функцию, то $f(\overline{x}, y)$ — уменьшенная на 1 степень y-ого простого, обозначим функцию, которая это делает:

$$f(\overline{x}, y) = \mathrm{ith}(y, m(\overline{x}, y)).$$

Вернемся к построению m:

$$\begin{cases} m(\overline{x}, 0) &= 2^{1+g(\overline{x})} \\ m(\overline{x}, y+1) &= m(\overline{x}, y) \cdot p_{y+1}^{1+h(\overline{x},\, y,\, f(\overline{x}, t_1(y)),\, \ldots,\, f(\overline{x}, t_s(y)))} \\ &= m(\overline{x}, y) \cdot p_{y+1}^{1+h(\overline{x},\, y,\, \mathrm{ith}(t_1(y), m(\overline{x}, t_1(y))),\, \ldots,\, \mathrm{ith}(t_s(y), m(\overline{x}, t_s(y))))} \end{cases}$$

[1] отрицательный корень можем сразу отбросить

Теорема 1.1.3 (Совместная рекурсия). Пусть $f_i^{(n+1)}$, $1 \leqslant i \leqslant k$,

$$\begin{cases} f_i(\overline{x}, 0) & = g_i(\overline{x}) \\ f_i(\overline{x}, y+1) & = h_i(\overline{x}, y, f_1(\overline{x}, y), \ldots f_k(\overline{x}, y)) \end{cases}$$

Если $g_i^{(n)}, h_i^{(k+2)}$, $1 \leqslant i \leqslant k$ — примитивно / общерекурсивные, то f_i тоже.

Основная идея этой теоремы — можем использовать y-е значение каждой из k функций.

□ Заметим, что канторовскую функцию можно, последовательно применив несколько раз, расширить до k-местной. Обозначим полученную функцию за c, а обратные за $c_1, \ldots c_k$.

Давайте просто объединим все f_i в одну функцию

$$m(\overline{x}, y) = c\left(f_1(\overline{x}, y), \ldots f_k(\overline{x}, y)\right).$$

Теперь каждую f_i можно вычислить

$$f_i(\overline{x}, y) = c_i(m(\overline{x}, y)).$$

Чтобы получить m достаточно использовать примитивную рекурсию:

$$\begin{cases} m(\overline{x}, 0) & = c\left(g_1(\overline{x}), \ldots g_k(\overline{x})\right) \\ m(\overline{x}, y+1) & = c(\\ & \quad h_1\left(\overline{x}, y, c_1(m(\overline{x}, y)), \ldots c_k(m(\overline{x}, y))\right), \\ & \quad\quad \vdots \\ & \quad h_k\left(\overline{x}, y, c_1(m(\overline{x}, y)), \ldots c_k(m(\overline{x}, y))\right) \\ &) \end{cases}$$

■

Теорема 1.1.4 (Кусочное задание функции). Пусть $R_0, \ldots R_k$ — отношения[a], такие что $\bigsqcup_{i=0}^{k} R_i = \mathbb{N}^m$ [b].

Для $|\overline{x}| = n$ кусочно зададим функцию $f^{(n)}$:

$$f(\overline{x}) = \begin{cases} f_0(\overline{x}), & \text{если } R_0(\overline{x}) \\ f_1(\overline{x}), & \text{если } R_1(\overline{x}) \\ \vdots & \vdots \\ f_k(\overline{x}), & \text{если } R_k(\overline{x}) \end{cases}$$

Если $f_i^{(n)}, R_i$ — примитивно / общерекурсивны, то и f тоже.

[a] Набор предикатов

[b] То есть для $i \neq j$ верно $R_i \cap R_j = \varnothing$.

□ Рассмотрим характеристические функции χ_{R_i} для R_i. Тогда

$$f(\overline{x}) = \sum_{i=0}^{k} f_i(\overline{x}) \cdot \chi_{R_i}(\overline{x}).$$

А это просто сумма произведений, которые мы можем вычислять. ■

1.2 Равносильность МТ и ЧРФ

Теорема 1.2.1. Функция вычисляется машиной Тьюринга тогда и только тогда, когда она частично рекурсивная (то есть вычислима по Клини).

▢

$\boxed{2 \Longrightarrow 1}$ Если $f(x_1, \dots x_n) = y$, то считаем, что МТ получаем $1^{x_1}01^{x_2}0\dots01^{x_n}$ и должна выдать 1^y; если f не определена, МТ должна зацикливаться и наоборот.

- Для простых функций можем построить МТ напрямую:
  - Если мы хотим выдавать нуль, просто стираем вход.
  - Если нужно увеличить число на один, приписываем 1 в конец справа.
  - Если нужно вернуть k-ую проекцию, стираем все до начала k-ого числа (то есть нужно отсчитать $k-1$ нуль на входе), далее стереть все после.
- Для операторов **S**, **R**, **M**:

  S: Пусть есть набор функций $h^{(n)}$, $g_1^{(m)}, \dots, g_n^{(m)} \longrightarrow f^{(m)}$, для каждой из которых есть машина Тьюринга M_h и M_{g_i}.
  Хотим построить МТ M_f для вычисления f.
  Сделаем это так:
  - Копируем весь вход n раз:
  $$\left(1^{x_1}01^{x_2}\dots01^{x_n}*\right)^n.$$
  - Запускаем M_{g_i} на соответствующей части полученного входа.
  Если нужно что-то записать, то будем сдвигать всю правую часть на нужное число клеток, чтобы освободить место.
  МТ запускаем псевдопараллельно (по очереди даем поработать).
  В каждой части после окончания работы оставляем только ответ:
  $$1^{y_1}*1^{y_2}\dots*1^{y_n},$$
  где $y_i = g_i(x_1, \dots x_m)$.
  - Запускаем на этом результате M_h.

  Лекция 2
  18 feb

  R: Пусть рекурсия задает $f^{(m+1)}(x_1, \dots x_m, y)$ из $g^{(m)}$ и $h^{(m+2)}$.
  $$\begin{cases} f(\overline{x}, 0) & = g(\overline{x}) \\ f(\overline{x}, y+1) & = h(\overline{x}, y, f(\overline{x}, y)) \end{cases}$$
  Считаем, что для g, h уже есть МТ (M_g и M_h), и мы хотим построить M_f, которая будет вычислять f.
  Построим вспомогательные МТ:
  - M_1: для входа $1^{x_1}0\dots01^{x_m}01^y$ построим $1^y01^{x_1}0\dots01^{x_m}0(1^0)01^{g(x_1,\dots x_m)}$. Для этого просто запустим M_g на входе, но не будем стирать его, а результат просто припишем после двух нулей справа. Далее мы будем накапливать значение u между этими нулями, а сейчас там ничего нет, то есть $u = 0$.
  - M_2: для входа $1^y01^{x_1}0\dots01^{x_m}01^u01^z$ построим $1^y01^{x_1}0\dots01^{x_m}01^{u+1}01^{h(x_1,\dots x_m,u,z)}$. Для этого, используя M_h, допишем в конец вместо z результат h и допишем единицу к 1^u. Здесь $u+1$ обозначает текущее значение y', а значение h — значение $f(y')$.
  - M_3: для входа $1^y01^{x_1}0\dots01^{x_m}01^u01^z$ оставим только 1^z.
  - Φ: для входа $1^y01^{x_1}0\dots01^{x_m}01^u01^z$ проверим, что $u \neq y$.

  Теперь соберем все вместе: сначала запустим M_1, далее пока Φ возвращает неравенство, запускаем M_2 (увеличиваем u на один, вычисляем следующее значение функции), и в конце стираем лишнее, запустив M_3.

M: Хотим по МТ M_g построить M_f, вычисляющую

$$f(\overline{x}) = \begin{cases} y & g(\overline{x}, y) = 0 \wedge g(\overline{x}, z) \neq 0 \quad \forall z < y \\ \uparrow & else \end{cases}$$

Аналогично построим несколько вспомогательных МТ:

– N_1 : приписывает 0 ко входу:

$$1^{x_1}0\dots01^{x_m} \longrightarrow 1^{x_1}0\dots01^{x_m}0.$$

– N_2 : дублирует вход, разделяя решеткой: $w \longrightarrow w\#w$

– N_3 : в продублированному входе меняет вторую половину на результат M_g

$$1^{x_1}0\dots01^{x_m}01^y\#1^{x_1}0\dots01^{x_m}01^y \xrightarrow{M_g} 1^{x_1}0\dots01^{x_m}01^y\#1^{g(x_1,\dots x_m,y)}.$$

– N_4 : очищает все после решетки и дописывает единицу в конец

$$1^{x_1}0\dots01^{x_m}01^y\#w \longrightarrow 1^{x_1}0\dots01^{x_m}01^{y+1}.$$

– N_5 : стирает все, кроме ответа

$$1^{x_1}0\dots01^{x_m}01^y\#w \longrightarrow 1^y.$$

– Φ : проверяет, что после решетки что-то еще есть $w\#v \longrightarrow v \neq \varepsilon$.

Теперь можем построить M_f так:

$$N_1;\ N_2;\ N_3; \text{while } \Phi \text{ do } N_4,\ N_2,\ N_3;\ N_5.$$

$\boxed{1 \Longrightarrow 2}$ Теперь мы хотим промоделировать работу МТ с помощью частично рекурсивной функции. На вход должны либо выдать результат, либо зациклится. Так как машины Тьюринга работают со строками, а функции с натуральными числами, нужно придумать правила кодирования.

Пусть есть конфигурация МТ

$$\alpha q_i a_j \beta,$$

где α — строка слева от головки, q_i — состояние, a_j — текущий символ, β — строка справа от головки.

Пронумеруем рабочий алфавит $\Gamma = \{a_0, \dots a_{m-1}\}$, где a_0 — пустой символ (␣).

Кодирование конфигураций Теперь можем конфигурацию записать как

$$\widetilde{\alpha}, \widetilde{q_i}, \widetilde{a_j}, \widetilde{\beta},$$

где $\widetilde{\alpha}$ — число, соответствующее α в m-ичной записи, $\widetilde{q_i}$ — просто номер состояния, $\widetilde{a_j}$ — номер в алфавите (j), $\widetilde{\beta}$ — число, соответствующее β в m-ичной записи, записанное справа налево.

Сдвиги обозначать будем d: вправо $d = 1$, влево $d = 2$.

Терминальное состояние — z. Множество состояний тоже пронумеруем и получим множество состояний $\widetilde{Q} = \{0, 1, \dots |Q| - 1\}$.

Пример 1.2.1

Рассмотрим небольшой пример. Пусть $\Gamma = \{a_0, a_1\}$, тогда следующее состояние будет записано как $(22, 3, 1, 13)$:

$$\underbrace{a_1a_0a_1a_1a_0}_{\alpha}\, q_3a_1\, \underbrace{a_1a_0a_1a_1}_{\beta}$$

Кодирование команд Пусть есть переход $(q,a) \rightarrow (p,b,d) = (p$ – новое состояние, b – новый символ, d – направление движения головки). Сопоставим p,b,d тройку функций $\varphi_q, \varphi_a, \varphi_d$:

$$\varphi_q\colon \widetilde{Q} \times \tilde{\Gamma} \to \widetilde{Q}$$
$$\varphi_a\colon \widetilde{Q} \times \tilde{\Gamma} \to \tilde{\Gamma}$$
$$\varphi_d\colon \widetilde{Q} \times \tilde{\Gamma} \to \{1,2\}$$

Эти функции будут примитивно рекурсивными, так как заданы на конечном множестве, на остальных можем доопределить нулем.

Преобразование конфигураций Пусть у нас есть переход между двумя конфигурациями:

$$K = \alpha q_i a_j \beta \to \alpha' q_i' a_j' b' = K'.$$

Зададим функцию на числах, которая проделает этот переход $\Phi\colon K \to K'$. На самом деле эта функция состоит из четырех, которые мы сейчас и определим.

Пусть

$$\begin{aligned}
\tilde{q_i}'(\tilde{\alpha},\tilde{q_i},\widetilde{a_j},\tilde{\beta}) &= \varphi_q(\tilde{q_i},\widetilde{a_j}) \\
\tilde{\alpha}'(\tilde{\alpha},\tilde{q_i},\widetilde{a_j},\tilde{\beta}) &= \begin{cases} \tilde{\alpha}\cdot m + \varphi_a(\tilde{q_i},\widetilde{a_j}), & \varphi_d(\tilde{q_i},\widetilde{a_j}) = 1 \\ \left\lfloor \frac{\tilde{\alpha}}{m} \right\rfloor, & \varphi_d(\tilde{q_i},\widetilde{a_j}) = 2 \end{cases} \\
\tilde{\beta}'(\tilde{\alpha},\tilde{q_i},\widetilde{a_j},\tilde{\beta}) &= \begin{cases} \tilde{\beta}\cdot m + \varphi_a(\tilde{q_i},\widetilde{a_j}), & \varphi_d(\tilde{q_i},\widetilde{a_j}) = 1 \\ \left\lfloor \frac{\tilde{\beta}}{m} \right\rfloor, & \varphi_d(\tilde{q_i},\widetilde{a_j}) = 2 \end{cases} \\
\widetilde{a_j}' &= \begin{cases} \tilde{\beta} \mod m, & \varphi_d(\tilde{q_i},\widetilde{a_j}) = 1 \\ \tilde{\alpha} \mod m, & \varphi_d(\tilde{q_i},\widetilde{a_j}) = 2 \end{cases}
\end{aligned}$$

Заметим, что все эти формулы примитивно рекурсивные[2].

Общая работа МТ Пусть $K(0) = \left(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0}\right)$ — начальная конфигурация. Чтобы получить новую конфигурацию для шага t, посчитаем все четыре параметра:

$$\begin{aligned}
K(t) = (& \\
& K_\alpha(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0},t) \\
& K_q(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0},t) \\
& K_a(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0},t) \\
& K_\beta(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0},t) \\
&)
\end{aligned}$$

Теперь запишем совместную рекурсию для $K_\alpha, K_q, K_a, K_\beta$:

$$\begin{cases} K_\alpha(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0},0) & = \widetilde{\alpha_0} \\ K_\alpha(\widetilde{\alpha_0},\widetilde{q_0},\widetilde{a_0},\widetilde{\beta_0},t+1) & = \tilde{\alpha}'\Big(K_\alpha(\ldots,t),K_q(\ldots,t),K_a(\ldots,t),K_\beta(\ldots,t)\Big) \end{cases}$$

Для остальных точно также.

Результат Пусть начальное состояние $q_0 a_0 \beta_0$ (стоим на самом левом символе), конечное — $q_z a_z \beta_z$, причем его встречаем впервые. То есть нам нужно вычислить функцию, которая переводит

$$x = \widetilde{a_0} + \widetilde{b_0} m \longrightarrow a_z + \beta_z m,$$

если машина Тьюринга пришла сюда, и не определена, если МТ зацикливается:

$$t_z = \mu t[K_q(t) = z].$$

[2]Единственное, чего нет явно в лемме 1 выше, это остаток по модулю, но его легко получить из деления нацело.

Тогда результатом работы МТ будет

$$\varphi(x) = m\cdot K_\beta\Big(0,\ 0,\ x \mod m,\ \lfloor \tfrac{x}{m} \rfloor,\ \mu t[K_q(0,\ 0,\ x \mod m,\ \lfloor \tfrac{x}{m} \rfloor, t) = z]\Big)$$
$$+K_a\Big(0,\ 0,\ x \mod m,\ \lfloor \tfrac{x}{m} \rfloor,\ \mu t[K_q(0,\ 0,\ x \mod m,\ \lfloor \tfrac{x}{m} \rfloor, t) = z]\Big)$$

■

Следствие 1. Любую частично рекурсивную функцию можно представить так, чтобы минимизация использовалась только один раз.

□ Сначала запишем для нее МТ, а потом постоим обратно функцию. В итоге получим эквивалентную функцию, причем по построению оператор минимизации использовался лишь один раз. ■

Следствие 2. Функция, вычислимая за примитивно рекурсивное время [a], тоже является примитивно рекурсивной.

[a]время, ограниченное примитивно рекурсивной функцией

□ В построении функции использовали минимизацию по числу шагов МТ, поэтому, если работаем примитивно рекурсивное время, можем применить ограниченную минимизацию. ■

1.2.1 Функция Аккермана

Можно построить общерекурсивную функцию, которая растет быстрее любой примитивно рекурсивной. Из этого следует, что **ПРФ** не совпадает с **ОРФ**.

Определение 8: Функция Аккермана

Функция Аккермана — функция от двух аргументов $\alpha_n(x)$, определенная следующим образом:

$$\begin{cases} \alpha_0(x) & = x+1 \\ \alpha_{n+1}(x) & = \alpha_n^{[x+2]}(x) = \underbrace{\alpha_n(\alpha_n(\ldots(x)))}_{x+2 \text{ раза}} \end{cases}$$

Лемма 3. $\alpha_n(x) \geqslant x+n+1$. В частности, $\alpha_n(x) > x$.

□ Докажем по индукции по n.

- База: $n = 0$. $\alpha_0(x) = x+1 = x+0+1$.
- Переход: $n-1 \to n$.

$$\begin{aligned} \alpha_n(x) &= \alpha_{n-1}^{[x+2]}(x) \geqslant \alpha_{n-1}^{[x+1]}(x)+1 > && \text{(т.к. } \alpha_{n-1}(t) > t \text{ по предположению индукции)} \\ &> \alpha_{n-1}^{[x]}(x)+1 > \alpha_{n-1}^{[x-1]} > \ldots > && \text{(продолжаем до кратности 1)} \\ &> \alpha_{n-1}(x)+1 \geqslant x+(n-1)+1+1 = x+n+1 \end{aligned}$$

■

Лемма 4. Если $x > y$, то $\alpha_n(x) > \alpha_n(y)$.

□ Индукция по n.

- База: $n = 0$. $\alpha_0(x) = x+1 > y+1 = \alpha_0(y)$.

- Переход: $n-1 \to n$.

$$\begin{aligned}
\alpha_n(x) &= \alpha_{n-1}^{[x+2]}(x) > \alpha_{n-1}^{[x+2]}(y) && \text{(по предположению для } n-1\text{)}\\
&> \alpha_{n_1}^{[x+1]}(y) > \ldots > \alpha_{n-1}^{[y+2]}(y) && \text{(по прошлой лемме 3)}\\
&= \alpha_n(y)
\end{aligned}$$

■

Лемма 5. Если $n > m$, то $\alpha_n(x) > \alpha_m(x)$.

□ Индукция по m.

- База: $m = 0$. По лемме 3 $\alpha_n(x) \geqslant x + n + 1 \geqslant x + 1 = \alpha_0(x)$.
- Переход: $m-1 \to m$. По определению $\alpha_n(x) = \alpha_{n-1}^{[x+1]}(x)$. Применим индукционное предположение ко всем α_{n-1} и заменим на α_{m-1}, после каждой замены значения будут уменьшаться:

$$\alpha_{n-1}^{[x+2]}(x) > \alpha_{n-1}^{[x+1]}(\alpha_{m-1}(x)) > \ldots > \alpha_{n-1}(\alpha_{m-1}^{[x+1]}(x)) > \alpha_{m-1}^{[x+2]}(x).$$

■

Лемма 6. Если $n > 1$, то $\alpha_n(x) > \alpha_{n-1}^{[2]}(x)$.

□ Очевидно по лемме 3. ■

Лемма 7. Для любой **ПРФ** $f(x_1, \ldots x_n)$ существует константа k, что $f(x_1, \ldots x_n) \leqslant \alpha_k(\max\{x_1, \ldots, x_n\})$. Если f имеет 0 аргументов, подставим 0.

□

- Для примитивных функций подойдет $k = 0$: для нульместного и одноместного нулей, $s(x)$ и проекции $I_a^b(x_1, \ldots, x_a)$ неравенство верно, так как $\alpha_0(x) = x + 1$.
- Если применяется оператор суперпозиции для других функций с найденными k_i, можно взять наибольшее из них (пусть k). Докажем, что подойдет $\alpha_{k+1}(x)$.

  Пусть суперпозиция применяется к $h(x_1, \ldots x_m)$ и $g_i(x_1, \ldots x_n)$.

$$\begin{aligned}
\alpha_{k+1}(\overline{x}) &\geqslant \alpha_k^{[2]} && \text{(лемма 7)}\\
&\geqslant \alpha_k\big(\max_{i\in[1,m]} g_i(\overline{x})\big) && \text{(лемма 6)}\\
&\geqslant h(g_1(\overline{x}), g_2(\overline{x}), \ldots g_m(\overline{x}))
\end{aligned}$$

- Если функция $f(\overline{x}, n)$ получена из $g(\overline{x})$ и $h(\overline{x}, n, t)$ с помощью примитивной рекурсии, сначала найдем k, чтобы $g(\overline{x}) \leqslant \alpha_k(\max \overline{x})$ и $h(\overline{x}, n, t) \leqslant \alpha_k(\max(\overline{x}, n, t))$.

  Оценим функцию f.

$$\begin{cases} f(\overline{x}, 0) = g(\overline{x}) \\ f(\overline{x}, n+1) = h(\overline{x}, n, f(\overline{x}, n)) \end{cases}$$

  Докажем, что $f(\overline{x}, n) \leqslant \alpha_k^{[n+1]}(\max(\overline{x}, n))$.[3]

  - База: $n = 0$. Верно, по определению f.

[3]Здесь под $\max(\overline{x}, \text{что-то еще})$ подразумевается максимум по всем координатам и n: $\max(x_1, \ldots, x_m, \text{что-то еще})$.

– Переход: $n-1 \to n$.

$$f(\overline{x}, n+1) \leqslant \alpha_k(\max(\overline{x}, n, f(\overline{x}, n))) \leqslant \alpha_k\Big(\max(\overline{x}, n, \alpha_k^{[n+1]}(\max(\overline{x}, n)))\Big)$$

(индукционное предположение)

$$= \alpha_k(\alpha_k(\max(\overline{x}, n))) = \alpha_k^{[n+2]}(\max(\overline{x}, n))$$

То есть

$$f(\overline{x}, n) \leqslant \alpha_k^{[n+1]}(\max(\overline{x}, n)) < \alpha_k^{[\max(\overline{x}, n)+2]}(\max(\overline{x}, n) = \alpha_{k+1}(\max(\overline{x}, n)).$$

■

Лемма 8. Для любой **ПРФ** $f(n)$ найдется такое $N \in \mathbb{N}$, что при $n > \mathbb{N}$ выполнено $\alpha_n(n) > f(n)$.

□ По лемме 7 для $f(n)+1$ найдется такое N, что $\alpha_N(n) \geqslant f(n)+1 > f(n)$. А тогда и для всех больших $m \geqslant N$ верно неравенство $\alpha_m(n) > f(n)$. ■

Теорема 1.2.2. $\alpha_n(n)\colon \mathbb{N} \to \mathbb{N}$ растет быстрее любой примитивно рекурсивной.

□ По лемме 8 для любой примитивно рекурсивной функции есть константа N, начиная с которой $\alpha_n(n) > f(n)$, то есть она будет расти быстрее.

Из этого также следует, что $\alpha_n(n)$ не **ПРФ**. ■

Лемма 9. $\alpha_n(x)$ является общерекурсивной функцией двух аргументов. В частности, $\alpha_n(n)$ — одного аргумента.

□ Построим машину Тьюринга A, вычисляющую $\alpha_n(x)$ по n и x: $x+2$ раза повторяем рекурсивно вызывать себя для $n-1$ и результата рекурсивных вызовов. Когда доходим до $n=0$, возвращаем число, увеличенное на один.

В итоге мы построили МТ строго по определению. ■

Chapter 2

Разрешимые и перечислимые множества

2.1 Определения

Определение 9: Разрешимое множество

Множество $X \subseteq \mathbb{N}^k$ называется ***разрешимым***, если его характеристическая функция вычислима[a].

[a] Это может быть частично рекурсивная функция, машина Тьюринга, λ-функция...

Замечание. Любое конечное множество разрешимо. Пересечение, объединение, разность разрешимых тоже разрешимо.

Теорема 2.1.1. Множество $X \subseteq \mathbb{N}$ разрешимо тогда и только тогда, когда X — множество значений всюду определенной вычислимой неубывающей функции (или пустое множество).

□

$\boxed{1 \implies 2}$ Можем в характеристической функции $\chi_X(n)$ возвращать n вместо 1, а в остальных значениях прошлое выданное. Эта функция подходит под описание.

$\boxed{2 \implies 1}$ Если множество конечно, то оно разрешимо, так как можем задать функцию χ_X на конечном числе точек. Если X бесконечно будем действовать, как описано далее.

Пусть есть функция f. Из нее хотим построить χ_X. Посчитаем $\chi_X(n)$ так: начнем с $i = 0$

- вычислим $f(i)$;
- если значение больше n, то в следующих входах, значения будут еще больше, поэтому можем сразу вернуть 0;
- если меньше, то посчитаем $f(i + 1)$ и вернемся к предыдущему пункту;
- так как функция неубывающая и достигает всех значений из X (причем их бесконечно много, поэтому есть элемент больше n), мы либо найдем значение больше n (тогда вернем 0), либо равное (тогда вернем единицу).

■

Лекция 3
25 feb

2.2 Перечислимые множества

Определение 10: Перечислимое множество

Множество $X \subseteq \mathbb{N}^k$ называется ***перечислимым***, если

- его *полухарактеристическая* функция вычислима:

$$\chi_X(n) = \begin{cases} 1, & n \in X \\ \uparrow, & n \notin X \end{cases}$$

- или, если существует алгоритм, который выводит все его элементы в некотором порядке.

Теорема 2.2.1 (Об эквивалентных определениях)**.** Следующие утверждения эквивалентны:

0. **Перечислимость:** существует алгоритм, который выводит все элементы X в некотором порядке
1. X — область определения вычислимой функции
2. X — область значений вычислимой функции
3. полухарактеристическая функция X вычислима
4. X — область значений всюду определена вычислимой функцией

▢

$\boxed{0 \Longrightarrow 3}$ Чтобы посчитать $\chi_X(n)$, запускаем алгоритм, перечисляющий элементы множества, ждем n.

Если вывелось n, то выводим $\chi_X(n) = 1$, а иначе мы зациклились, то есть получили расходимость.

$\boxed{3 \Longrightarrow 1}$ Действительно, множество X будет областью определения χ_X, а она вычислима.

$\boxed{1 \Longrightarrow 0}$ Пусть область определения вычисляется алгоритмом B, который просто считает функцию, если вход принадлежит области определения, алгоритм завершается, иначе зацикливается.

Построим алгоритм A следующим образом: будем запускать B по шагам и выводить элементы множества

- 1 шаг на входе 0
- 2 шага на 0, 2 шага на 1
- 3 шага на $0, 1, 2$
- и так далее
- как только B закончил работу на некотором элементе, выводим этот его.

Этот алгоритм A перечисляет наше множество, так как для алгоритма B требуется конечное время работы на элементах области определения.

$\boxed{2 \Longrightarrow 0}$ Аналогично, но выводим значение функции на элементе, на котором мы останавливаемся.

$\boxed{1 \Longrightarrow 2}$ Пусть X — область определения функции, которая вычисляется алгоритмом A. Рассмотрим следующую функцию:

$$b(n) = \begin{cases} n, & \text{если } A \text{ заканчивает работать на } n \\ \uparrow, & \text{если } A \text{ зацикливается на } n \end{cases}$$

Теперь X — область значений $b(n)$, а она вычислима.

$\boxed{0 \Longrightarrow 4}$ Пусть A — алгоритм, перечисляющий X. Рассмотрим любой $n_0 \in X$.

Построим функцию f, которая всюду определена и X — ее область значений.

$$f(n) = \begin{cases} t, & \text{если на } n\text{-ом шаге работы } A \text{ появляется } t \\ n_0, & \text{если ничего не появляется} \end{cases}$$

$\boxed{4 \Longrightarrow 2}$ Очевидно

■

Замечание. Все области значений и определений не применимы к пустому множеству, которое тоже перечислимое.

Задача. В определении перечислимого множества можно выводить элементы с повторениями. Это эквивалентно определению без повторений.

Теорема 2.2.2. Если считать перечислимыми только множества, для которых существует машина Тьюринга, выводящая каждый элемент множества *ровно по разу*, то их класс не поменяется.

□ Увеличиться класс точно не может, так как мы накладываем более строгое условие.

Проверим, что по обычной МТ M, перечисляющей множество A, можно построить МТ M', которая будет выводить все элементы ровно один раз.

Пусть машина M' работает почти как, но записывает на ленту все числа, которые она уже выводила.

Теперь, если M должна вывести число, M' проверяет, что еще не возвращала его ранее, записывает и выдает. Если число уже записано, возвращать не будем. ■

Теорема 2.2.3. Объединение и пересечение перечислимых множеств тоже перечислимое.

□ По определению перечислимости для первого множества есть алгоритм A, который завершается на всех элементах этого множества. Аналогично для второго — B.

- Хотим проверить, что n принадлежит объединению. Будем давать алгоритмам A и B поработать по шагу. Ждем шага, на котором завершает работу хотя бы один алгоритм. Значит, n лежит в одном из множеств.

- Чтобы проверить принадлежность пересечению запустим сначала A, если он завершит работу, то запустим B. Если и B остановится, n лежит в обоих множествах.

Если элемент не принадлежит объединению или пересечению, получим расходимость. ■

Теорема 2.2.4 (Пост). A разрешимо тогда и только тогда, когда A и $\overline{A}$ перечислимые.

□

$\boxed{1 \Longrightarrow 2}$ Так как A разрешимо, можем рассмотреть вычислимую характеристическую функцию $\chi_A(n)$.

Построим полухрарактеристические для A и $\overline{A}$:

- для A: если $n \in A$, то $\chi'_A = 1$, иначе $\chi'_A(n)$ расходится
- для $\overline{A}$ аналогично, только результаты инвертированы.

$\boxed{2 \Longrightarrow 1}$ Пусть мы хотим проверить $n \overset{?}{\in} A$.

Запускаем одновременно по шагам алгоритмы, перечисляющие A и $\overline{A}$, ждем появления n. Рано или поздно должно появиться, так как в объединении A и $\overline{A}$ дают все множество.

Если его выдал алгоритм для A, то $\chi_A(n) = 1$, а если для $\overline{A}$, то $\chi_A(n) = 0$.

■

Определение 11: Проекция

Подмножество $P \subseteq \mathbb{N}$ называется ***проекцией*** $Q \subseteq \mathbb{N}^2$, если

$$\forall x \colon x \in P \iff \exists y \colon (x, y) \in Q.$$

Теорема 2.2.5 (О проекции)**.** Множество P перечислимое тогда и только тогда, когда P — проекция некоторого разрешимого множества Q.

□

$\boxed{1 \implies 2}$ Пусть A — алгоритм, перечисляющий P. Тогда подойдет

$$Q \coloneqq \{(n, t) \mid n \text{ появляется в течение } t \text{ шагов работы } A\}.$$

$\boxed{2 \implies 1}$ Проекция перечислимого перечислима: берем алгоритм, которые перечисляет Q, но оставляем только первую координату.

■

Теорема 2.2.6 (О графике)**.** Частичная функция $f \colon \mathbb{N} \to \mathbb{N}$ вычислима тогда и только тогда, когда перечислим ее график

$$F \coloneqq \{(x, y) \mid f(x) \text{ определена и } f(x) = y\}.$$

□

$\boxed{1 \implies 2}$ Чтобы перечислить все все точки графика будем по очереди запускать алгоритм для f на $x \in \mathbb{N}$, причем будем давать ему поработать i шагов на шаге i.

Если $f(x)$ в некоторый момент вычислено, выводим $(x, f(x))$.

$\boxed{2 \implies 1}$ Построим алгоритм вычисляющий f.

Пусть нам на вход дан элемент x, запустим алгоритм, перечисляющий элементы F.

Если $(x, f(x)) \in F$, то есть f определена в точке x, и в какой-то момент мы выпишем значение.

Если же функция не определена в x, то пары $(x, f(x))$ в F нет и мы зацикливаемся.

■

Замечание. Различные названия типов множеств в других источниках:

перечислимое	разрешимое
полуразрешимое	
вычислимо перечислимое	вычислимое
полурекурсивное	рекурсивное
semi-decidable	decidable
semi-recursive	recursive
enumerable	

2.3 Универсальные функции

Определение 12

Сечением функции $U^{(m+1)}(n, \overline{x})$ назовем функцию $U_n(\overline{x})$ от m аргументов, которая получается из U

фиксацией первого аргумента.

Определение 13: Универсальная функция

$U(n, \overline{x})$ — ***универсальная*** для класса K функция от m аргументов, если

- $\forall n\colon U_n(\overline{x}) \in K$
- $\forall f \in K\ \exists n\colon f = U_n$

То есть множество ее сечений совпадает с K.

Замечание. Универсальная функция существует только для счетных K.

Замечание. Все рассматриваемые функции частичные.

Обозначение. *$\mathcal{F}^m$ — вычислимые функции от m аргументов.*

$\mathcal{F}_*^m$ — всюду определенные вычислимые функции от m аргументов.

Без верхнего индекса по умолчанию подразумевается единица.

Теорема 2.3.1. Существует вычислимая функция 2-х аргументов $U \in \mathcal{F}^2$, универсальная для класса вычислимых функций 1-ого аргумента $\mathcal{F}^1$.

□ Запишем все коды МТ, вычисляющих функции из $\mathcal{F}^1$, в порядке возрастания (сначала по длине, затем в алфавитном).

Пусть $U(i, x)$ — функция, которая находит запись i-ой МТ M_i, запускает ее на входе x и возвращает результат.

Во-первых, U вычислима, так как вычисляется описанным выше алгоритмом.

Во-вторых, сечение U_i соответствует МТ M_i, поэтому U универсальна для $\mathcal{F}^1$. ■

Следствие 3. Существует $U' \in \mathcal{F}^{m+1}$, универсальная для $\mathcal{F}^m$.

Замечание. Здесь мы будем использовать m-местную канторовскую нумерацию $c(x_1, \dots, x_m)$, которую можно построить, например, последовательным сворачиванием пар. Обозначим обратные проекции на i координату $c_i(y) = x_i$.

□ Проверим, что универсальной функцией будет

$$U'(n, \overline{x}) \coloneqq U(n, c(\overline{x})),$$

где U — универсальная для $\mathcal{F}^1$.

Во-первых, заметим, что все сечения вычислимы.

Далее рассмотрим произвольную функцию $f(\overline{x}) \in \mathcal{F}^m$. Найдем для нее одно из сечений U'.

Определим

$$g(y) \coloneqq f\Big(c_1(y), \dots c_m(y)\Big).$$

g вычислимая, U универсальная, поэтому

$$\exists n\colon U_n(y) = g(y).$$

$$\begin{aligned}
U'(n, \overline{x}) &= && \text{(по определению } U\text{)}\\
= U(n, c(\overline{x})) &= && (n \text{ – номер } g)\\
= g(c(\overline{x})) &= && \text{(по определению } g\text{)}\\
= f(c_1(c(\overline{x})), \dots c_m(c(\overline{x}))) &= f(\overline{x})
\end{aligned}$$

То есть U' действительно универсальная. ■

Теорема 2.3.2. Не существует $U \in \mathcal{F}_*^2$ универсальной для $\mathcal{F}_*$.

☐ Предположим, что такая функция $U \in \mathcal{F}_*^2$ существует.

Рассмотрим диагональную функцию:

$$d'(n) = U(n,n) + 1 \in \mathcal{F}_*.$$

С одной стороны, $d'(n)$ — общерекурсивная функция, поэтому из универсальности U следует, что существует сечение $U_n = d'$.

С другой стороны, $d'(n)$ отличается от всех сечений U: если $\forall x\colon U(n,x) = d'(x)$, подставим $x = n$, получим $U(n,n) = U(n,n) + 1$. Противоречие. ■

Замечание. Для класса частичных функций такое рассуждение не проходит, так как они могут быть не определены и прибавление единицы ничего не меняет для неопределенности.

Теорема 2.3.3. Существует вычислимая частичная функция, которая не имеет всюду определенного вычислимого продолжения [a].

[a]то есть нельзя доопределить до всюду определенной вычислимой

☐ Подходит функция $d'(n) = U(n,n) + 1$, где U — универсальная вычислимая функция [1].
Пусть ее можно доопределить до вычислимой d'':

$$d'' = \begin{cases} U(n,n) + 1, & \text{такие } n, \text{ где } U(n,n) \text{ определена} \\ \text{определена}, & \text{где } U(n,n) \text{ не определена} \end{cases}$$

Поэтому d'' отличается от всех сечений универсальной функции. Противоречие. ■

2.3.1 Перечислимое неразрешимое множество

Аналогично можно ввести определения для перечислимых множеств.

Определение 14: Сечение множества

Сечением множества $W \subseteq \mathbb{N}^k$ назовем $W_n = \{\overline{x} \mid (n, \overline{x}) \in W\}$.

Докажем аналог прошлой теоремы для множеств.

Теорема 2.3.4. Существует перечислимое множество $W^{(m+1)} \subset \mathbb{N}^{m+1}$, являющееся универсальным для всех перечислимых подмножеств $\mathbb{N}^m$.

☐ Рассмотрим универсальную функцию $U^{(m+1)}$. Пусть множество $W^{(m+1)}$ — ее область определения, оно будет перечислимым.

Пусть у нас есть перечислимое множество $X \in \mathbb{N}^m$.

Найдем такую функцию $f \in \mathcal{F}^m$, для которой X — область определения, и такое n, что $U_n = f$.

Тогда $W_n = X$. ■

Теорема 2.3.5. Существует перечислимое неразрешимое множество.

☐ Рассмотрим вычислимую $f(x)$, не имеющую всюду определенного вычислимого продолжения.

Пусть F — ее область определения, она перечислима и неразрешима:

[1]далее это обозначение по умолчанию определяет универсальную вычислимую частичную функцию $U^{(m+1)}$ для вычислимых частичных функций m аргументов

- F перечислимо, потому что оно является областью определения вычислимой функции;
- F неразрешимо, так как в противном случае можно рассмотреть общерекурсивное доопределение f:

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

Эта функция всюду определена, вычислима, является продолжением f. Противоречие.

Замечание. $F = \{n \mid U(n,n) \text{ определено}\}$ — переформулировка класса L_1 (останавливающиеся на своем входе МТ).

Замечание. $\overline{F}$ — пример неперечислимого множества.

Замечание. Область определения универсальной функции перечислимое, но не разрешимое множество, так как область определения $U(n,n)$ — частично определенная неразрешимая.

Замечание. «Проблема остановки»[2] — переформулировка принадлежности данной функции к области определения универсальной функции.

■

Теорема 2.3.6. Существует частичная вычислимая функция $f\colon \mathbb{N} \to \{0,1\}$, которая не имеет всюду определенного вычислимого продолжения.

□ Определим

$$d'''(n) = \begin{cases} 1, & U(n,n) = 0 \\ 0, & U(n,n) > 0 \\ \uparrow, & U(n,n) \uparrow \end{cases}$$

Любое доопределение будет отличаться от $U(n,n)$, так как для всех n значения $d'''(x)$ и $U_n(x)$ различны: здесь либо обе функции расходятся, либо первое не равно второму. ■

Определение 15

Непересекающиеся множества называются ***отделимыми разрешимым***, если существует разрешимое множество, содержащее одно из них и непересекающееся с другим.

Следствие 4. Существуют перечислимые непересекающееся неотделимые никаким разрешимым множеством множества.

□ Подойдут следующие множества: $X = \{n \mid d'''(n) = 0\}$ и $Y = \{n \mid d'''(n) = 1\}$ Пусть существует разделяющее их разрешимое C, содержащее Y и непересекающееся с X.

Тогда χ_C — общерекурсивное дополнение d'''. Противоречие. ■ Лекция 4 4 march

2.3.2 Главные универсальные функции

Определение 16

Пусть $U^{(n+1)} \in \mathcal{F}^{n+1}$ универсальная нумерация для функций $\mathcal{F}^n$.
U называется ***главной нумерацией*** или ***главной универсальной функцией***, если для любой нумерации $V \in \mathcal{F}^{n+1}$ существует ***транслятор*** $s \in \mathcal{F}_*$[a], такой что

$$\forall m \in \mathbb{N},\ \overline{x} \in \mathbb{N}^n\colon \quad V(m,\overline{x}) = U(s(m),\overline{x}).$$

[2]останавливается ли МТ M на входе x

[a]по номеру сечения V находит какой-то номер такого же сечения U

Теорема 2.3.7. Существует главная универсальная функция $U^{(n)} \in \mathcal{F}^{n+1}$ для класса $\mathcal{F}^n$.

▢ По следствию 3 существует $T \in \mathcal{F}^{n+2}$, универсальная для $\mathcal{F}^{(n+1)}$. Построим U:

- Определим $U(x, \overline{y}) \coloneqq T(l(x), r(x), \overline{y})$.

  Здесь, как обычно, c — канторовская нумерация, l, r — левая и правая обратные функции.

- Докажем, что U главная. Пусть $V \in \mathcal{F}^{n+1}$ — некоторая функция. Так как T универсальная для содержащего V класса, существует m (номер функции V среди сечений T), такой что

  $$\forall x,\ \overline{y}: \quad V(x, \overline{y}) = T(m, x, \overline{y}).$$

  Проверим, что транслятор $s(x) = c(m, x)$ подойдет, то есть $V(x, \overline{y}) = U(s(x), \overline{y})$.

  $$\begin{aligned} &U(s(x), \overline{y}) && = && \text{(По определению } U\text{)} \\ &= T(l(s(x)), r(s(x)), \overline{y}) && = T(l(c(m,x)), r(c(m,x)), \overline{y}) = && \\ &= T(m, x, \overline{y}) && = V(x, \overline{y}) && \end{aligned}$$

  Значит, U главная.

■

Второе доказательство:
▢ Аналогично мы строили универсальную функцию выше (теорема 2.3.1 и следствие 3). Она и будет главной.[3] ■

Теорема 2.3.8 (О вычислимости номера композиции)**.** $U \in \mathcal{F}^2$ — универсальная функция для класса $\mathcal{F}$. U — *главная* универсальная тогда и только тогда, когда существует $f \in \mathcal{F}^2_*$, такая что

$$U_p \circ U_q = U_{f(p,q)}.$$

То есть $\forall p, q, x \in \mathbb{N}: \quad U(p, U(q, x)) = U(f(p,q), x)$.

▢

$\boxed{1 \implies 2}$ Пусть U — главная.

Рассмотрим $V(n, x) = U(l(n), U(r(n), x))$, то есть $V(c(p,q), x) = U(p, U(q, x))$.

Фактически V — это $U_q \circ U_p$. Так как U — главная универсальная,

$$\exists s \in \mathcal{F}_*: \quad V(n, x) = U(s(n), x).$$

Тогда

$$\begin{aligned} U_p \circ U_q = U(p, U(q,x)) = V(c(p,q), x) = \\ = U(s(c(p,q)), x) = U_{s(c(p,q))}(x) \end{aligned} \qquad \text{(По определению транслятора)}$$

Теперь обозначим $f(p,q) = s(c(p,q))$ и получим нужное равенство.

$\boxed{2 \implies 1}$

▢ Пусть есть такая функция $f(p,q) \in \mathcal{F}^2_*$, что $U_p \circ U_q = U_{f(p,q)}$. Хотим доказать, что U главная универсальная.

[3]Здесь можно было бы и расписать подробно.

- Найдем транслятор для канторовской нумерации $t \in \mathcal{F}$: $\forall n, x \in \mathbb{N} \quad U(t(n), x) = c(n, x)$.
  Рассмотрим две вспомогательные функции

$$k(z) = c(0, z)$$
$$g(z) = c(l(z) + 1, r(z))$$

  Так как U универсальная, эти функции имеют номера, пусть n_k и n_g соответственно.
  Тогда в качестве транслятора можно взять следующую функцию:

$$\begin{cases} t(0) = n_k \\ t(n+1) = f(n_g, t(n)) \end{cases}$$

  Проверим это по индукции по n.
  - База: $n = 0$. $U(t(0), x) = U(n_k, x) = U_{n_k}(x) = k(x) = c(0, x)$, что и требовалось.
  - Переход: $n \to n + 1$.

$$\begin{aligned} U(t(n+1), x) = U(f(n_g, t(n)), x) &= U_{n_g} \circ U_{t(n)}(x) && \text{(определение } t \text{ и свойство } f\text{)} \\ &= g \circ c_n(x) = g(c(n, x)) && \text{(предположение индукции)} \\ &= c(n+1, x) \end{aligned}$$

  Теперь можем пользоваться t.
- Построим транслятор для любой функции $V \in \mathcal{F}^2$. Пусть $g(y) = V(l(y), r(y))$. Так как U универсальная, есть сечение $U_a = g$. Подставим $c(n, x)$:

$$U_a(c(n, x)) = g(c(n, x)) = V_n(x).$$

  Выразим $c(n, x)$ через U с помощью транслятора t:

$$\begin{array}{lcl} U_a(U(t(n), x)) & = U_a \circ U_{t(n)}(x) = V_n(x) \\ \shortparallel & & \\ U_{f(a, t(n))}(x) & \qquad\qquad = V_n(x) \end{array}$$

Тогда $s(n) = f(a, t(n))$ — нужный транслятор для V.

2.3.3 Теорема Райса

Определение 17: Свойство функций

Свойство $\mathcal{A}$ функций класса $\mathcal{C}$ — подмножество функций, удовлетворяющих этому свойству, то есть лежащих в $\mathcal{A}$.

Нетривиальное свойство — не пустое и не совпадающее со всем классом: $\varnothing \subsetneq \mathcal{A} \subsetneq \mathcal{C}$.

Теорема 2.3.9 (Райса / Успенского). Пусть $\mathcal{A} \subset \mathcal{F}$ — некоторое нетривиальное свойство вычислимой функции, U — главная универсальная функция для всех вычислимых функций класса $\mathcal{F}$.

Тогда не существует алгоритма, который по U-номеру вычислимой функции проверяет $\mathcal{A}$.

То есть множество $A = \{n \mid U_n \in \mathcal{A}\}$ неразрешимо.

□ Покажем, что, если свойство $\mathcal{A}$ можно алгоритмически проверить, то любые два непересекающихся перечислимых множества можно отделить некоторым разрешимым.

Пусть P и Q — произвольные непересекающиеся множества.

ξ — какая-нибудь функция из $\mathcal{A}$, а η — какая-нибудь не из $\mathcal{A}$.

Рассмотрим следующую функцию:

$$V(n,x) = \begin{cases} \xi(x), & x \in P \land x \notin Q \\ \eta(x), & x \in Q \land x \notin P \\ \uparrow, & n \notin P \cup Q \end{cases}$$

Заметим, что V вычислимая, так как можем запустить по шагам алгоритмы для P и Q, если один из них завершается, то остается вычислить соответствующую функцию, а иначе значение не определено.

Так как V вычислима, а U универсальна, существует транслятор $s \in \mathcal{F}_*$:

$$U(s(n),x) = V(n,x).$$

Теперь для любого с помощью проверки $V_n(x) \in \mathcal{A}$ можем отделить P от Q. Получаем противоречие со следствием 4. ∎

Следствие 5. Множество номеров некоторой заданной функции φ в главной нумерации неразрешимо.

В частности, в главной нумерации множество МТ, вычисляющих одну функцию, бесконечно много.

Следствие 6 (Пример универсальной неглавной функции)**.** Существует универсальная неглавная для класса $\mathcal{F}^n$ функция $V \in \mathcal{F}^{n+1}$.

□ Пусть $U(n,x)$ — произвольная главная универсальная функция для $\mathcal{F}^n$ и D — множество номеров функций в нумерации U с непустой областью определения.

Заметим, что D перечислимое, так как можно построить следующий алгоритм: на шаге k будем для $i \in \{0,1,\ldots,k-1\}$ и $\bar{j} \in \{0,1,\ldots,k-1\}^n$ считать $U_i(\bar{j})$. Для этого даем на каждой из k^{n+1} машин Тьюринга (для $U_i(\bar{j})$) поработать k шагов. Теперь для всех пар $(i,\bar{j})$, на которых был выдан результат, выводим i.

Для любой функции U_i с непустой областью определения рано или поздно найдется k, которое больше номера функции и координат какой-то точки из области определения и количество действий для вычисления значения в ней, так как мы, во-первых, мы перебираем все точки, а, во-вторых, постоянно увеличиваем количество шагов.

И когда мы дойдем до этого k номер i будет выведен. Поэтому алгоритм действительно перечисляет номера функций, но, естественно, только с непустой областью определения.

Определим функцию $f(n)$, для n равную n-ому элементу D в порядке возвращения построенным алгоритмом.

Можно заметить, что по теореме Райса, D неразрешимо (свойство «иметь непустую область определения» очевидно непустое, и D — множество номеров таких функций).

Поэтому D бесконечно, следовательно, для всех n когда-то будет выведен n-ый элемент D.

Теперь рассмотрим функцию

$$V(i,\overline{x}) = \begin{cases} \uparrow, & i = 0 \\ U(f(i-1),\overline{x}), & i \neq 0 \end{cases}$$

Эта функция вычислима, универсальна. При этом единственный номер «нигде не определенного сечения» — только 0, это множество конечно, следовательно разрешимо. Поэтому V неглавная по теореме Райса.

Также любая где-то определенная функция будет получена для какого-то V_n, поэтому V универсальна. ∎

Следствие 7 (Переформулировка прошлого следствия)**.** Для любой главной нумерации U и любой вычислимой функции f множество $\{n \mid U_n = f\}$ неразрешимо.

2.4 Теорема о неподвижной точке

Лемма 10. Пусть $\equiv$ — отношение эквивалентности на $\mathbb{N}$.
Тогда следующие утверждения *не* выполняются одновременно:

1. Для любой $f \in \mathcal{F}$ существует $\equiv$-*продолжение* $g \in \mathcal{F}_*$[a].
2. Найдется $h \in \mathcal{F}_*$, не имеющая $\equiv$-неподвижной точки, то есть $\forall n\colon n \not\equiv h(n)$.

[a]То есть, если $f(x)$ определена, то $g(x)$ тоже определена и $g(x) \equiv f(x)$

□ Рассмотрим $f \in \mathcal{F}$, от которой никакая вычислимая функция не может отличаться всюду, например, $f(x) = U(x, x)$.

Пусть выполняются оба пункта.

1. По первому существует $\equiv$-продолжение f функция $g \in \mathcal{F}_*$.
2. По второму существует такая $h \in \mathcal{F}_*$, что $\forall n\colon h(n) \not\equiv n$.

Рассмотрим $t(x) \coloneqq h(g(x))$ и проверим, что она всюду отличается от f:

- Если f определена, то $f(x) \equiv g(x) \not\equiv h(g(x)) = t(x)$
- Если f не определена, то t определена

Но от f никакая вычислимая функция не может отличаться всюду. ■

Теорема 2.4.1 (О неподвижной точке)**.** Если U — главная универсальная вычислимая функция для класса $\mathcal{F}$, а $h \in \mathcal{F}_*$, то $\exists n\colon U_n = U_{h(n)}$.

□ Возьмем в качестве отношения эквивалентности следующее: $x \equiv x \iff U_x = U_y$.

Покажем, что выполняется первый пункт из леммы 10.

Пусть $f \in \mathcal{F}$. Тогда можем рассмотреть $V(n, x) \coloneqq U(f(u), x)$.

Так как U главная существует транслятор:

$$\exists s \in \mathcal{F}_*\colon \forall n, x\ V(n, x) = U(s(n), x).$$

Проверим, что s и есть $\equiv$-продолжение f

- если $f(n)$ определена, то $U_{s(n)} = U_{f(n)}$, то есть $s(n) = f(n)$.
- если не определена, то и $U_{s(n)}$ нигде не определена.

В итоге первый пункт леммы выполняется, поэтому второй не выполняется. ■

Следствие 8. $U(n, x)$ — главная универсальная вычислимая функция. Тогда

$$\exists p \in \mathbb{N}\colon \quad \forall x\ U(p, x) = p.$$

□ Рассмотрим $V \in \mathcal{F}^2$, такую что $V(n, x) = n$. Тогда существует $s(n)$, такое что $U_{s(n)} = V_n = n$.

Теперь применим теорему о неподвижной точке к $s(n)$:

$$\exists p\colon \quad U_p = U_{s(p)} = V_p = p.$$

■

2.5 m-сводимость

Определение 18: m**-сводимость**
Множество $A \subset \mathbb{N}$ m***-сводится*** $(A \leqslant_m B)$ к $B \subset \mathbb{N}$, если существует $f \in \mathcal{F}_*$, такая что

$$\forall x \in \mathbb{N}: \quad x \in A \Longleftrightarrow f(x) \in B.$$

Свойства.

- Если $A \leqslant_m B$ и B разрешимо, то A разрешимо.
- Если $A \leqslant_m B$ и B перечислимо, то A перечислимо.
- Отношение $\leqslant_m$ рефлексивно и транзитивно.
- Если $A \leqslant_m B$, то $\mathbb{N} \setminus A \leqslant_m \mathbb{N} \setminus B$.

□

- Чтобы проверить $x \overset{?}{\in} A$, проверим $f(x) \in B$. Так как B разрешимо, вторая проверка выдаст какой-то ответ и мы можем его вернуть.
- Аналогично, если $f(x) \in B$, то $x \in A$, а если расходится, то $x \notin A$.
- Для рефлексивности подойдет $f = \mathrm{id}$, для транзитивности берем композицию.
- Инвертируем результат:

$$x \in \mathbb{N} \setminus A \Longleftrightarrow x \notin A \Longleftrightarrow f(x) \notin B \Longleftrightarrow f(x) \in \mathbb{N} \setminus B.$$

■

Замечание. Разрешимое множество сводится к любому $B \notin \{\infty, \mathbb{N}\}$

Замечание. К пустому множеству сводится только пустое. к $\mathbb{N}$ сводится только $\mathbb{N}$.

Определение 19: m**-полнота**
Перечислимое множество A называется m***-полным*** (в классе перечислимых множеств), если любое перечислимое B m-сводится к A.

Лекция 5
11 march

Теорема 2.5.1. Существует m-полное перечислимое множество.

□ Подойдет любое универсальное множество $U \subset \mathbb{N} \times \mathbb{N}$.

Пусть $V \subset \mathbb{N}$ — множество номеров пар из U:

$$c(x, y) \in V \Longleftrightarrow (x, y) \in U.$$

Проверим, что это множество подходит.

Путь T — произвольное перечислимое множество, так как U универсальное,

$$\exists n: \quad T = U_n.$$

Тогда

$$x \in T \Longleftrightarrow x \in U_n \Longleftrightarrow (n, x) \in U \Longleftrightarrow c(n, x) \in V.$$

То есть $f(x) = c(n, x)$ сводит T к V. ■

Замечание. Если K — m-полное, $K \leqslant_m A$ — перечислимое, то A тоже m-полное.

Теорема 2.5.2. $U \subset \mathbb{N} \times \mathbb{N}$ — главное универсальное перечислимое множество. Тогда его диагональ $D = \{x \mid (x, x) \in U\}$ будет m-полной.

□ Во-первых, D перечислимое. Далее предположим, что K — произвольное перечислимое множество. Тогда $V = K \times \mathbb{N}$ перечислимо.

$$V_n = \begin{cases} \varnothing, & n \notin K \\ \mathbb{N}, & n \in K \end{cases}.$$

Так как U главная

$$\exists s \in \mathcal{F}_* : V_n = U_{s(n)} = \begin{cases} N, & n \in K \\ \varnothing, & n \notin K \end{cases}$$

Тогда

$$s(n) \in U_{s(n)} \iff s(n) \in D \iff n \in K.$$

То есть $s(n)$ сводит K к D. ■

Утверждение. Существует перечислимые неразрешимые множества, которые не являются m-полными в классе перечислимых.

□ Пока без доказательства, возможно будет в будущем на лекции. ■

2.6 Проблема соответствия Поста (PCP)

Это одна из самых известных неразрешимых задач.

Даны два конечных списка строк одинаковой мощности над алфавитом A :

$$\begin{aligned} L_1 &: w_1, \dots w_k \quad x_i, w_i \in A^* \\ L_2 &: x_1, \dots x_k \quad |L_1| = |L_2| \end{aligned}$$

Хотим проверить, существуют ли конечные последовательности $i_1, \dots i_m \in \{1, \dots, k\}^+$, такие что $w_{i_1} \dots w_{i_m} = x_{i_1} \dots x_{i_m}$.

Пример 2.6.1

Пусть $L_1 = a^2, b^2, ab^2$, $L_2 = a^2b, ba, b$. Решением будет 1213:

$$w_1 w_2 w_1 w_3 = aabbaaabb = x_1 x_2 x_1 x_3.$$

Пример 2.6.2

Теперь возьмем $L_1 = a^2b, a$, $L_2 = a^2, ba^2$. Несложный перебор приводит к тому, что решений нет.

Переформулировки

- Через гомоморфизмы. Даны два гомоморфизма $h_1, h_2 : \Delta^* \to A^*$. Проверяем, если ли строка $u \in \Delta^* : h_1(u) = h_2(u)$
- Через доминошки. Есть k типов доминошек (w_i, x_i), каждого типа бесконечно много. Проверяем, можно ли составить последовательно доминошки, чтобы срока сверху совпала со строкой снизу.

Теорема 2.6.1 (Пост, 1946)**.** Проблема соответствия Поста неразрешима.

▢ Сведем к задаче об остановке односторонней одноленточной МТ. Отметим начало ленты символом $\triangleright$, $\_$— вместо ε. Начальное состояние q_0 не входит в правые части команд.

По МТ $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{term}$ построим пример для задачи Поста. Считаем, что в конце M стирает все.

Хотим записать историю вычисление МТ.

Пусть $A = \{\triangleright, \#, \Gamma, Q\}$, $\#$ — для разделения конфигураций.

1. Начальные доминошки $(\varepsilon, \triangleright q_0 x\#)$, $x \in \Sigma \cup \_$
2. Доминошки копирования (c, c), $c \in A$
3. Для всех правил $(q, c) \to (q', c', +1)$: $(qc, c'q')$, если $c = \_$, добавляем еще доминошку $(q\#, c'q'\#)$.
4. Для всех правил $(q, c) \to (q', c', -1)$: $(aqc, q'ac')$, если $c = \_$, добавляем $(aq\#, q'ac'\#)$
5. Конец строки, для всех $c \in A \setminus \triangleright$, три доминошки $(cq_{term}, q_{term}), (q_{term}, c), (\triangleright q_{term} q_{term}, q_{term})$

Замечание. Доминошки копирования дают решение задачи Поста, далее это поправим, а пока считаем, что начинаем с доминошки типа 1.

Шаг 1 Сверху пусто, снизу начальная конфигурация.

Шаг 2 Обязательно доминошка $(\triangleright, \triangleright)$

Шаг i Сверху $i-1$ конфигурация МТ, снизу i-ая. Доминошка копирования, доминошка команды, доминошка копиорования

Если МТ не останавливается, то последовательность доминошек не совпадет.

Если МТ останавливается, то достаточно добавить доминошку 5 в конец.

Добьемся того, чтобы можно было начинать с доминошки копирования. Например, можно сделать две копии алфавита A и A', дальше раздвоить каждую доминошку: (сверху A, снизу A'), (A', A). Теперь, если поставить на первое место копирование, то работа МТ также моделируется. ■

Следствие 9. Задача о пустом пересечении 2-х грамматик алгоритмически неразрешима.

▢ Пусть разрешима. Тогда PCP разрешима следующим образом:

$$\begin{aligned} Gr_1 &: S \to x_i S w_i^R / \# \quad \forall i \in \{1, \dots k\} \\ Gr_2 &: S \to aSa / a\#a \qquad\quad \forall a \in A \end{aligned}$$

То есть Gr_1 — строка вида $x_{i_1} \dots x_{i_n} \# (w_{i_1} \dots w_{i_m})^R$, а Gr_2 — $v\#v^R$, где $v \in A^+$

$$L(Gr_1) \cap L(Gr_2) \neq \varnothing \iff \text{PCP имеет решение.}$$

■

Замечание. PCP неразрешима даже для бинарного алфавита, так как можем взять инъективное кодирование бинарными словами.

Замечание. Можно доказать, что PCP неразрешима для списков длины $k = 5$ (без доказательства)

Замечание. Для $k = 2$ разрешима, для $k \in \{3, 4\}$ неизвестно.

2.7 T-сводимость (по Тьюрингу)

Определение 20: Сводимость по Тьюрингу

Пусть $A, B \subset \mathbb{N}$. Тогда B ***сводится по Тьюрингу*** к A ($B \leqslant_T A$), если существует алгоритм с аракулом A, отвечающим на вопрос о принадлежности n множеству B.

Свойства.

- $B \leqslant_m A \implies B \leqslant_T A$
- $\forall A\colon A \leqslant_T \mathbb{N} \setminus A$
- сводимость по тьюрингу транзитивна и рефлексивна
- $A \leqslant_T B$ и B разрешимо, то A тоже разрешимо

Замечание. Если A перечислимо, $B \leqslant_T A$, то не обязательно B перечислимо.

Определение 21: Функция с оракулом

Аналогично можно определить вычисления с оракулом f (это всюду определенная функция). Функция вычисляется алгоритмом, который в любой момент мжет обратиться к оракулу — попростить оракула вычислить $f(n)$.

Лекция 6
18 march

2.8 Арифметическая иерархия

Вспомним следующее свойство: $A \subset \mathbb{N}$ перечислимо, согда $\exists B \subset \mathbb{N} \times \mathbb{N}$, такое что A — проекция B, или

$$x \in A \iff \exists y\ (x, y) \in B \tag{2.8.1}$$

Можем считать A, B свойствами (предикатами), то есть $A(x) \hookrightarrow x \in A$. Тогда можем переписать 2.8.1 так

$$A(x) \iff \exists y\ B(x, y).$$

Какие множества представимы в виде $\forall y\colon B(x, y)$? Это равносильно

$$\neg\left(\exists y\ \neg B(x, y)\right).$$

Это ***коперечислимые*** (то есть дополнение до перечислимого)

Определение 22

$A \in \Sigma_n$, если его можно представить в виде

$$A(x) \iff \exists y_1 \forall y_2 \exists y_3 \ldots B(x, y_1, \ldots, y_n),$$

где B разрешимо.

$A \in \Pi_n$, если

$$A(x) \iff \forall y_1 \exists y_2 \forall y_3 \ldots B(x, y_1, \ldots, y_n).$$

Свойства.

1. Определение не изменится, если разрешить несколько одинаковых кванторов подряд, так как можем заменить повторные на один с помощью канторовой нумерации
2. $A(x) \in \Sigma_n \iff \neg A(x) \in \Pi_n$

Теорема 2.8.1. Если $A(x), B(x) \in \Sigma_n$ (или Π_n), то

$$A(x) \cup B(x) \in \Sigma_n \text{ (или } \Pi_n)$$

$$A(x) \cap B(x) \in \Sigma_n \text{ (или } \Pi_n).$$

☐ Запишем определения

$$A(x) \iff \exists y_1 \forall y_2 \exists y_3 \dots P(x, y_1, \dots y_n)$$
$$B(x) \iff \exists z_1 \forall z_2 \exists z_3 \dots Q(x, z_1, \dots z_n)$$

Скомбинируем кванторы

$$A(x) \cap B(x) \iff \underbrace{\exists y_1 \exists z_1}_{\exists c(y_1, z_1)} \forall y_2 \forall z_2 \dots \underbrace{P(x, y_1, \dots y_n) \cap Q(x, z_1, \dots z_n)}_{l(c(y,z))}.$$

Так как P и Q разрешимы их объединения и пересечения тоже разрешимы. ■

Замечание. Аналогично можно определить Σ_n, Π_n для подмножеств $\mathbb{N}^m$.

Свойства.

- $\Sigma_n,\ \Pi_n \subseteq \Sigma_{n+1},\ \Pi_{n+1}$
- $\Sigma_n \cup \Pi_n \subseteq \Sigma_{n+1} \cap \Pi_{n+1}$

☐ Аналогично прошлой теореме сворачиваем кванторы в группы, добавляем кванторы. ■

В итоге получается следующая картина

$$\begin{array}{ccccccccc} \Sigma_0 & \subseteq & \Sigma_1 & \subseteq & \Sigma_2 & \subseteq & \Sigma_3 & \subseteq & \dots \\ \| & & \rotatebox{45}{$\subseteq$} & & \rotatebox{-45}{$\subseteq$} & & & & \\ \Pi_0 & \subseteq & \Pi_1 & \subseteq & \Pi_2 & \subseteq & \Pi_3 & \subseteq & \dots \end{array}$$

Теорема 2.8.2. $A \leqslant_m B,\ B \in \Sigma_n$, то $A \in \Sigma_n$ (аналогично с Π_n)

☐ Распишем согласно определениям

$$A \leqslant_m B \overset{\text{def}}{\iff} \exists f \in \mathcal{F}_* \colon x \in A \iff f(x) \in B$$

и

$$B \in \Sigma_n \overset{\text{def}}{\iff} x \in B \iff \exists y_1 \forall y_2 \dots R(x, y_1, y_2, \dots y_n).$$

Тогда

$$x \in A \iff f(x) \in B \iff \exists y_1 \forall y_2 \dots R(f(x), y_1, \dots y_n).$$

Так как f вычислима, то и $R(f(x), y_1, \dots y_2)$ разрешимо. ■

Теорема 2.8.3. Для любого $n > 0$ в классе Σ_n (соответственно Π_n) существует множество, универсальное для всех множеств в Σ_n (соответственно Π_n)

Замечание. Если A -универсальное в Σ_n, то $\overline{A}$ — универсальное в Π_n

Определение 23: Универсальное множество
Множество $W \subset \mathbb{N} \times \mathbb{N}$ ***универсальное*** для перечислимого подмножества $\mathbb{N}$, если

- W перечислимое;
- для всех перечислимых $B \subset \mathbb{N}$ существует номер n,такой что $W_n = B$.

Определение 24: Главное универсальное множество
Множество $W \subset \mathbb{N} \times \mathbb{N}$ ***главное универсальное*** для всех перечислимых подмножеств $\mathbb{N}$, если для любого $V \in \mathbb{N} \times \mathbb{N}$ существует функция $s\colon \mathbb{N} \to \mathbb{N}$:

- s вычислима и всюду определена;
- $\forall x, n\colon (n, x) \in V \iff (s(n), x) \in W$

Например, таким будет область определения универсальной функции.

☐ Докажем по индукции.
Для Σ_1 – перечислимое, уже строили. Для Π_2

$$\forall y \underbrace{\exists z \underbrace{R(x, y, z)}_{\text{разрешимо}}}_{P(x,y)} \Longrightarrow \forall y \underbrace{P(x, y)}_{\text{перечислимое свойство}}.$$

Рассмотрим $U(n, x, y)$ — универсальное множество для перечислимых. Тогда

$$T(n, x) \coloneqq \forall y U(n, x, y).$$

$T(n, x)$ получается универсальным для Π_2.
Следовательно, существует универсальное для Σ_2 — дополнение до $T(n, x)$.
Продолжаем далее по индукции: для 3 начинаем с Σ_3

$$\Sigma_3\colon \exists y \forall z \exists t R(x, y, z, t) \iff \exists y \forall z P(x, y, z).$$

Универсальное для Σ_3:

$$T(n, x) = \exists y \forall z U(n, x, y, z), \text{где } U \text{ — универсальное перечислимое.}$$

И так далее ■

Теорема 2.8.4. Универсальное множество для Σ_n не принадлежит Π_n и наоборот.

☐ Пусть $T(m, x)$ — универсальное Σ_n-свойство. И предположим, что $T(m, x) \in \Pi_n$.
Рассмотрим $D(x) = T(x, x) \in \Pi_n$, так как $D \leqslant_m T$.
Поэтому $\neg D(x) \in \Sigma_n$, но оно отличается от всех сечений $T(n, x)$. Противоречие. ■

Следствие 10. $\Sigma_n \subsetneq \Sigma_{n+1}$ и $\Pi_n \subsetneq \Pi_{n+1}$.

Цель: доказать, что $\forall n\ \exists A \subset \mathbb{N}$, такое что $\Sigma_n = \{X \mid X\ A\text{-перечислимо}[4]\}$

2.9 Еще про T-сводимость

Зафиксируем некоторую функцию-оракула α. Вся теория вычислимости может быть «релятивизована» относительно вычислений с оракулом α.

Определение 25

Функцию, вычислимую с оракулом α будем называть α-вычислимой.

Обозначение. *F_α^m — класс α-вычислимых функций от m аргументов.*

Теорема 2.9.1. Пусть α — всюду определенная функция. Тогда $\exists U_\alpha(n, x) \in \mathcal{F}_\alpha^2$ — универсальная для класса $\mathcal{F}_\alpha^1$

□ Пусть $U_\alpha(i, x)$ — результат применения i-ой МТ с оракулом α к x. Как устроен оракул нам не важно, поэтому можем просто вшить обращение к нему во входные данные. ■

Аналогично перечислимым множествам можем определить α-перечислимые множества как:

- область определения α-вычислимой функции
- область значений α-вычислимой функции
- проекция α-разрешимого множества

Теорема 2.9.2.

1. Для любого $X \subset \mathbb{N}$ существует универсальное X-перечислимое множество для X-перечислимых.
2. Это множество будет m-полным в классе X-перечислимых.

□ Доказательство полностью аналогично такой же теореме для обычной перечислимости. ■

Определение 26

Множества P и Q являются ***m-эквивалентными*** ($P \equiv_m Q$), если $P \leqslant_m Q$ и $Q \leqslant_m P$.
m-степень — $\deg_m(P) = \{Q \mid Q \equiv_m P\}$.

Определение 27

Множества P и Q являются ***T-эквивалентными*** ($P \equiv_T Q$), если $P \leqslant_T Q$ и $Q \leqslant_T P$.
T-степень — $\deg_T(P) = \{Q \mid Q \equiv_T P\}$.

Замечание. Если $P, Q \in \deg_T(X)$, то $\mathcal{F}_P = \mathcal{F}_Q$ и P-перечислимость эквивалентна Q-перечислимости.

Поэтому можем говорить о $\deg_T$ X-неперечислимости.

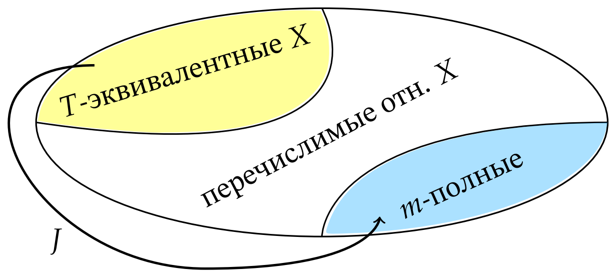
Определение 28: Операция скачка

Операция скачка J: $\deg_T \to \deg_T$ (или $\deg_m \to \deg_m$) выбирает для X какое-то m-полное относительно X-перечислимости.
То есть по всем эквивалентным $\deg_T(X)$ получаем X-перечислимые и среди них выбираем полные.

T-степени

- $\mathbb{O}$ — все разршимые.
- $\mathbb{O}' = J(\mathbb{O})$ — степень m-полного перечислимого неразрешимого. Один из представителей — ООП универсальной функции.
- $\mathbb{O}^{(n+1)} = \left(\mathbb{O}^{(n)}\right)'$

Можно считать, что $\mathbb{O}'$ — разрешимые с оракулом проблемы остановки.



2.10 Теорема об арифметической иерархии

Теорема 2.10.1 (Об арифметической иерархии). $\forall n \geqslant 1\colon \Sigma_n = \{\mathbb{O}^{(n-1)}$ — перечислимые множества$\}$

□ Для $n = 1$ уже знаем.

$\boxed{\subset}$ Рассмотрим $X \in \Sigma_2$, тогда

$$x \in X \iff \exists y \forall z \underbrace{R(x,y,z)}_{\text{разрешимо}}.$$

Навешиваем отрицание

$$\neg\big(\forall z R(x,y,z)\big) \in \Sigma_1.$$

Принадлежность Σ_1 дает m-сводимость к m-полному перечислимому множество. А так как m-сводимость влечет T-сводимость, по определению $\mathbb{O}'$ множество $\mathbb{O}'$-разрешимо.

Следовательно, его проекция $\mathbb{O}'$-перечислима.

Аналогично действуем для больших n:

$$x \in \Sigma_n\colon x \in X \iff \exists y \underbrace{R(x,y)}_{\in \Pi_2}$$

Тогда $\neg R \in \Sigma_{n-1}$.

На предыдущем слое доказали, что тогда $\neg R$ будет $\mathbb{O}^{(n-2)}$-перечислимо, а тогда он $\mathbb{O}^{(n-1)}$-разрешимо. Тогда его проекция $\mathbb{O}^{(n-1)}$-перечислима.

$\boxed{\supset}$ См. далее (страница 38).

■

2.10.1 Утверждения для доказательства в обратную сторону

Определение 29

Рассмотрим некоторую вычислимую нумерацию конченых множество.

Пусть D_x — множество с номером x.

Возьмем $A \subset \mathbb{N}$ (не обязательно конченое).

Определим $\mathrm{Subset}(A) = \{x \mid D_x \subset A\}$ — множество номеров конечных подмножество A.

Аналогично $\mathrm{Disjoint}(A) = \{x \mid D_x \cap A = \varnothing\}$.

Лемма 11 (о Subset). Если $A \in \Sigma_n$ (или Π_n), то $\mathrm{Subset}(A) \in \Sigma_n$ (или Π_n соответственно).

□ Пусть $A \in \Sigma_3$,

$$x \in A \iff \exists y \forall z \exists t \underbrace{T(x, y, z, t)}_{\text{разрешимо}}.$$

Для конечного набора

$$\{x_1, \dots x_m\} \subset A \iff \exists (y_1, \dots y_m) \forall (z_1, \dots, z_m) \exists (t_1, \dots t_m) \bigwedge_{i=1}^{m} R(x_i, y_i, z_i, t_i).$$

А это равносильно $c(x_1, \dots x_m) \in \mathrm{Subset}(A)$. ■

Лемма 12 (о Disjoint)**.** Если $A \in \Sigma_n$ (или Π_n), то Disjoint $\in \Pi_n$ (или Σ_n соответственно).

□

$$D_x \cap A = \varnothing \iff D_x \in \overline{A} \iff x \in \mathrm{Subset}(\overline{A})$$

То есть $\mathrm{Disjoint}(A) = \mathrm{Subset}(\overline{A})$. Тогда

$$A \in \Sigma_n \iff \overline{A} \in \Pi_n \implies \qquad \text{(по лемме о Subset)}$$
$$\mathrm{Disjoint}(A) \in \Pi_n$$

■

Относительная вычислимость: эквивалентные определения

Определение 30: Образец

Образец — функция $\mathbb{N} \to \mathbb{N}$, определенная на конечном множестве значений. Задается конченым множеством пар, то етсь можем вычислимо пронумеровать образцы.

Образцы ***совместимы***, если объединение их графиков есть график функции.

Определение 31

M — множество троек (x, y, t), где $x, y \in \mathbb{N}$, а t — образец.

Две тройки $(x_1, y_1, t_1), (x_2, y_2, t_2)$ ***противоречат друг другу***, если $x_1 = x_2, y_1 \neq y_2$ и t_1 и t_2 совместны.

Мноежство M ***корректно***, если оно не содержит противоречащих троек.

Определение 32

Пусть M — корректное множество троек, α — функция.

$$M_1 = \{(x, y, t) \mid (x, y, t) \in M, t \text{ — подмножество графика } \alpha\}$$
$$M_2 = \{(x, y) \mid \exists t\colon (x, y, t) \in M_1\}$$

Тогда M_2 определяет график некоторой функции $M[\alpha]$. Причем определение корректно, так как для всех x не больше одного значения.

Лекция 7
25 march

Теорема 2.10.2. Частичная функция $f \in \mathcal{F}_\alpha^{(1)}$, где $\alpha\colon \mathbb{N} \to \mathbb{N}$ всюду определена, равносильно тому, что существует корректное множество троек M, такое что $f = M[\alpha]$.

□

$\boxed{1 \implies 2}$ У нас есть алгоритм с оракулом α, вычисляющий f. Построим M.

Построим дерево, моделирующее работу алгоритма на входе n по всем значениям оракула. Скорее всего, дерево будет бесконечным, с бесконечным числом веток.

Если на входе x возможен ответ y, запишем тройку (x, y, t) в M, где t — образец, содержащий все ответы оракула на этой ветке.

- M перечислимо, так как можем стандартным образом давать поработать по k первым входам k шагов, а k увеличивать от 0 до ∞.
- M корректно. Пусть $(x, y_1, t_1), (x, y_2, t_2) \in M$ и $y_1 \neq y_2$. Эти тройки соответствуют разным путям в дереве (так как $y_1 \neq y_2$), найдем вершину, где они разделились. В этом месте рассматриваются два разных ответа оракула, причем первый содержится в t_1, а второй — в t_2, поэтому t_1 и t_2 несовместны.

  Поэтому противоречивых троек нет.

Следовательно, M корректно.

Проверим, что $f = M[\alpha]$. Пусть $f(x) = y$. Это соответствует ветке дерева, начинающейся с входа x и заканчивающееся y, возможно с запросами к α.

Рассмотрим t — образец, содержащий пары вопрос-ответ в этой ветке. t является частью α, а $(x, y, t) \in M$.

Значит, $M[\alpha](x)$ определено и равно y.

$\boxed{2 \Longrightarrow 1}$ Пусть есть корректное перечислимое множество троек M, такое что $f = M[\alpha]$.

Нужно построить алгоритм с оракулом α, считающий функцию f.

На входе x запускаем алгоритм, перечисляющий M. Выбирает тройку, в которой первый элемент равен x.

Обозначим тройку (x, y, t). Так как t конечное, по каждому элементу можем задать вопрос оракулу α, тем самым проверим, что t является частью α.

Если да, то $f(x) := y$, иначе продолжаем спрашивать про следующий элемент. Если «да» никогда не получаем, то функция не определена в этой точке.

В итоге мы построили алгоритм, который вычисляет $M[\alpha]$.

∎

Определение 33

Пусть α — перечислимое множество. Рассмотрим произвольное множество E пар (x, t), где $x \in \mathbb{N}$ и t — образец.

$$E[\alpha] := \{x \mid \exists (x, t) \in E,\ t \text{ является частью } \alpha\}.$$

Теорема 2.10.3 (о характеризации относительной вычислимости). X — α-перечислимое тогда и только тогда, когда существует перечислимое E, такое что $X = E[\alpha]$.

□

$\boxed{1 \Longrightarrow 2}$ Если X — α-перечислимое, то X — область определения α-вычислимой функции f. По предыдущей теореме $f = M[\alpha]$, для некоторого перечислимого корректного M.

Можем получить E выкалыванием второй координаты из M.

Так как $E[\alpha]$ является областью определения $M[\alpha] = f$, $E[\alpha] = X$.

$\boxed{2 \Longrightarrow 1}$ Пусть $X = E[\alpha]$ для некоторого E.

Рассмотрим $M = \{(x, 0, t) \mid (x, t) \in E\}$, оно корректно, поэтому соответствующая функция $M[\alpha]$ будет определена на $X = E[\alpha]$ и принимает только значение 0.

■

Продолжим доказательство главной теоремы (страница 35).

□ Сначала докажем, что $\mathbb{O}'$ перечислимо и лежит в Σ_2.

Пусть A является $\mathbb{O}'$-перечислимым. По определению A перечислимо относительно некоторого перечислимого B. То есть оно перечислимо относительно ξ_B.

По теореме о характеристике относительной вычислимости, существует перечислимое множество Q пар вида (x, t), где $x \in \mathbb{N}$ и t — образец, такое что

$$x \in A \iff \exists t\colon (x, t) \in Q, \xi_B \text{ продолжает } t.$$

Можем считать, что

- «ξ_B продолжает t» означает, что B содержит множество, на котором $t = 1$, и не пересекается с множеством, на котором $t = 0$
- функция, соответствующая t, принимает только 0 и 1, так как иначе ξ_B не сможет ее продолжить

Поэтому вместо *образцов* в данном случае можно рассматривать *пары конечных множество* и вместо *множества пар* Q — *множество* P *троек* (x, u, v), где

$$x \in A \iff \exists u \exists v \Big((x, u, v) \in P \wedge \underbrace{D_u \subset B}_{\substack{\Updownarrow \\ u \in \mathrm{Subset}(B)}} \wedge \underbrace{D_v \cap B = \varnothing}_{\substack{\Updownarrow \\ v \in \mathrm{Disjoint}(B)}} \Big).$$

Так как P перечислимо, $P \in \Sigma_1$, второе свойство ($u \in \mathrm{Subset}(B)$) по лемме о Subset принадлежит Σ_1, а третье $v \in \mathrm{Disjoint}(B)$ принадлежит Π_1 по лемме о Disjoint.

То есть все условие в скобках принадлежит Σ_2, поэтому и вся правая часть из Σ_2. Доказали для $n = 2$.

Дальше действуем аналогично, заменив 2 на n. ■

Следствие 11. $\Sigma_n \cap \Pi_n = \{\mathbb{O}^{(n-1)}\text{-разрешимые}\}$

□ Релятивизованная теорема Поста. ■

Следствие 12. $\Sigma_n \cup \Pi_n \subsetneq \Sigma_{n+1} \cap \Pi_{n+1}$ для $n > 0$

□ По определению $\mathbb{O}^{(n)}$ это степень m-полного множества X в классе $\mathbb{O}^{(n-1)}$-перечислимых.

Если X является m-полным, то не разрешимо в этом классе, поэтому $\overline{X}$ не является перечислимым в этом классе.

По теореме об арифметической иерархии $x \in \Sigma_n$, $\overline{X} \in \Pi_n$ и $\overline{X} \notin \Sigma_n$.

Рассмотрим $A = \{2n \mid n \in X\} \cup \{2n + 1 \mid n \notin X\}$. Так как к A сводится и X, и $\overline{X}$, $A \notin \Sigma_n$ и $A \notin \Pi_n$. Значит, $A \in \Sigma_{n+1} \cap \Pi_{n+1}$. ■

2.11 Классификация множеств в иерархии

Теорема 2.11.1. Множество номеров нигде не определенной функции в главной нумерации Π_1-полное

□ Упражнение. Подсказка: его дополнение Σ_1-полное ■

Теорема 2.11.2. 1. Если $U \in \mathcal{F}^2$ — универсальное для $\mathcal{F}$. Тогда

$$\{n \mid U_n \text{ всюду определено тождественно равным } 0\} \in \Pi_2$$

2. Если U — главная, то это множество Π_2-полное.

□

1. Пусть
$$A = \{n \mid \forall k\ \exists t\ U(n,k) \text{ заканчивает работу за } t \text{ шагов и выдает } 0\}.$$
Это Π_2.

2. Докажем, что к нему сводится произвольное множество из из Π_2, то есть
$$x \in P \iff \forall y\ \exists z\ \underbrace{R(x,y,z)}_{\text{разрешимо}}.$$
Рассмотрим $S(x,y)$, которая перебирает z и ищет такое, что $R(x,y,z)$ выполнено. Если нашли, возвращаем 0.

   То есть $S_x \equiv 0 \iff x \in P$.

   Так как U главная нумерация, $\exists s\colon U_{s(x)} = S_x$. То есть s сводит P к множеству номеров, которое тождественно нулевой функции.

■

Пример на третьем слое — множества с конечными дополнениями.

Задача. Пусть f — вычислимая функция, U — главная нумерация. $A = \{n \mid U_n = f\}$.

Найти минимальный класс для множества A (он зависит от функции).

Chapter 3

Дополнительная лекция

Эта лекция читалась только студентам программ “Математика” и “Науки о данных” 12 марта 2021 [1]. Лекция 8 12 march

3.1 Замощения плитками Ванга

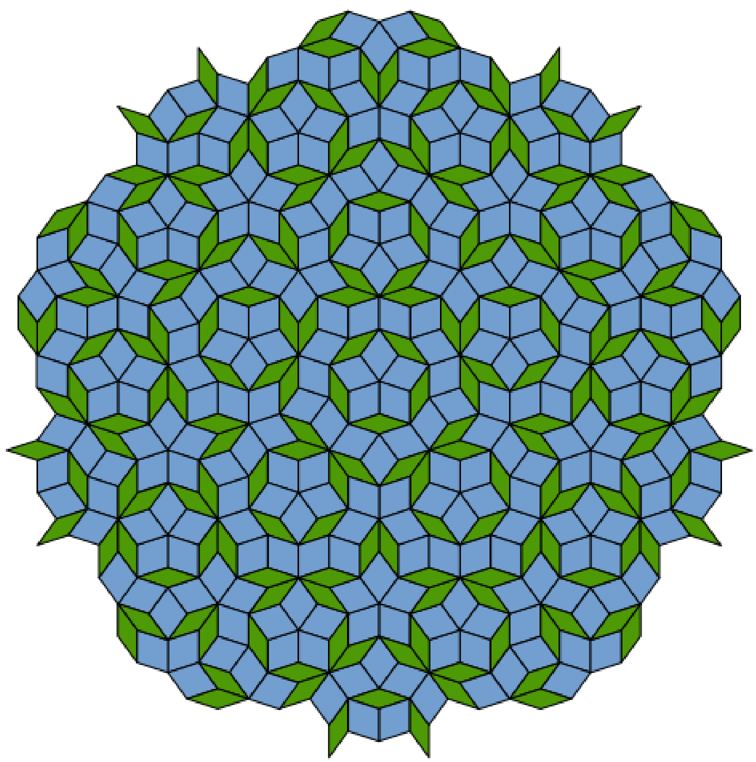


Figure 3.1: Замощение Пенроуза

Замещение Пенроуза не периодично, то есть *апериодично*, но покрывает всю плоскость.

Основная наша цель – доказать, что если нам дан на вход набор 11-и плиток Ванга, то нельзя выяснить, можно ли ими замостить плоскость (то есть, эта задача неразрешима).

Задача формулируется так – даются плиточки 1×1, мы их можем помещать в целочисленные точки плоскости. Каждую из сторон мы можем помечать, например, цветами – скажем, левую и верхнюю сторону – красной, правую – фиолетовой, нижнюю – желтой. Ставить рядом можно только стороны одинакового цвета.

[1] Набирал Даниил Любаев, кратинки и правки – Вячеслав Тамарин

Определение 34: Замощение

Замощение — это отображение $t\colon \mathbb{Z}^2 \to T$, где T – набор плиток.

Замечание. Самих плиток бесконечно, но количество их типов – конечно.

Пример 3.1.1

Если плитка типа 1 – справа и сверху синий цвет, а снизу и слева – красный, а плитка типа 2 – наоборот, то получится только шахматное замощение.

Вопрос, который задавал Ванг (1961): *если дан на вход набор плиток, существует ли алгоритм, проверяющий существование замощения?*

Вопрос решил его студент Berger в 1966 году, доказав неразрешимость. Мы докажем упрощенный вариант – замощение *с зерном.*

Теорема 3.1.1. Задача замощения с зерном (то есть, выделенная плитка $s \in T$ обязана присутствовать в замощении) алгоритмически неразрешима.

☐ Сведем задачу остановки машины Тьюринга на пустой ленте.

Дана $M = (Q, \Sigma, \Gamma, , \delta, q_0, f)$, где f – конечно. Пробел обозначаем за B (blank).

Строим T:

1. Плитки инициализации.

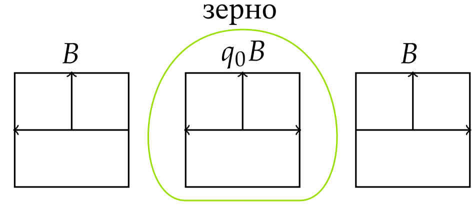


Figure 3.2: Плитки инициализации

2. Плитки алфавита

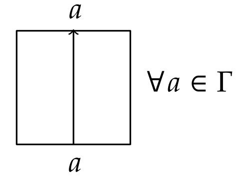


Figure 3.3: Плитки алфавита

3. Плитки переходов. Если есть команда $\delta(q, a) = (p, b, t)$, то сопоставляем ей плитку (1).

   Если команда $\delta(q, a) = (p, b, \to)$, то плитка (2).

4. Плитки склеивания.

5. Пустая плитка, чтобы заполнить низ.

Посмотрим теперь, как плитки устроены и как мы будем делать сведение.

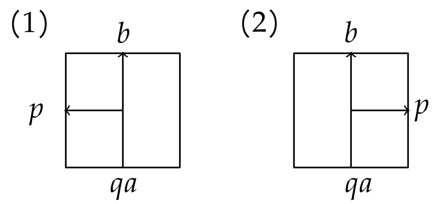


Figure 3.4: Плитки перехода

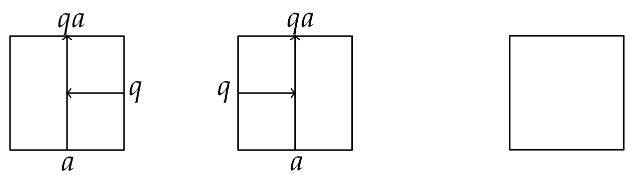
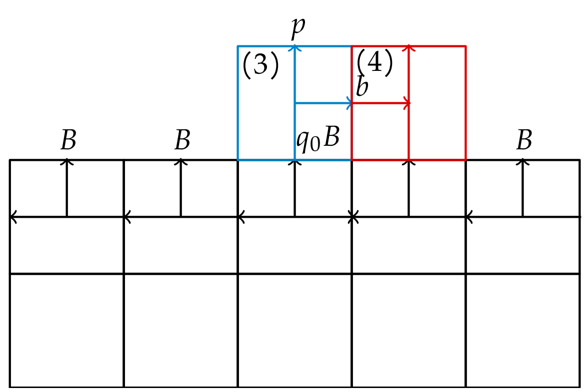


Figure 3.5: Плитки склеивания и пустая плитка

Начинаем с зерна (потому что оно должно присутствовать). Что мы можем к ней приписать справа? Только одну из плиток инициализации (какую, видно из картинки). Слева – аналогично.

Какую можем вниз? Только плитку типа 5.

Какую можем сделать при шаге машины Тьюринга? Выглядит это примерно так:



Все остальное – плитки склеивания, вариантов нет.

Продолжается это бесконечно, потому что если машина остановится, то это будет конечное число шагов, и мы не сможем прилепить плитку (потому что переход будет не определен).

Метки бесконечных строк составляют конфигурацию машины Тьюринга. Замощения существуют тогда и только тогда, когда машина Тьюринга не останавливается. Получается, если бы мы умели решать задачу замощения, то мы бы могли сказать, остановится ли машина Тьюринга. ∎

Лемма 13. Если для любого n существует замощение квадрата $n \times n$, то существует замощение всей плоскости.

□ Аналог леммы Кенига о том, что в бесконечном дереве существует бесконечная ветвь. Или так: любая плитка встречается бесконечно много раз. Смотрим ее возмозные продолжения до квадратика 2×2. Какой-то из этих вариантов встречается бесконечно много раз – выберем его. Смотрим ее продолжения до квадратика 3×3, и т.д. ∎

Следствие 13. Неразрешимость в купе с леммой дает существование апериодических замощений, то есть наборов плиток, для которых существуют только непериодические замощения.

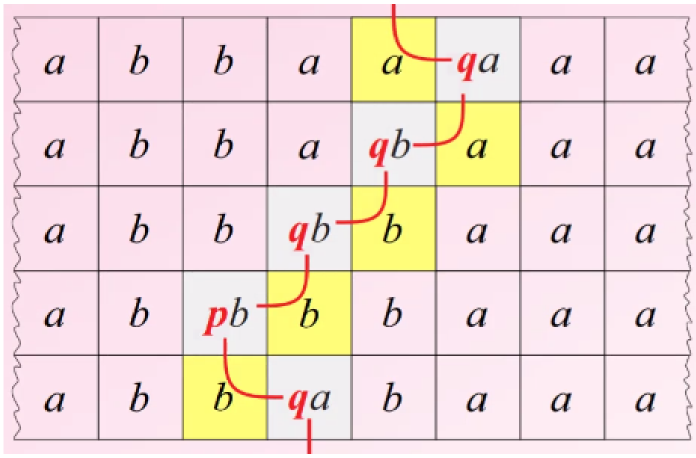


Figure 3.6: Имитация МТ

□ Иначе можем красить увеличивающиеся квадраты $n \times n$, либо придем к противоречию, либо увидим период. ■

Лемма 14. Если для данного набора плиток существует замощение, периодическое в одном направлении, то существует и замощение, периодическое в двух направлениях.

□ Пусть (a, b) – вектор периодичности (то есть, если сдвигаем плитку на вектор (a, b) то там та же плитка).

Рассмотрим кусочки такого размера *картинка*

Квадратов бесконечно, способов замостить конечно, поэтому какой-то встретится два раза. При этом цвета снизу такие же, как и сверху (потому что период). Значит красным квадратом мы можем замостить все. ■

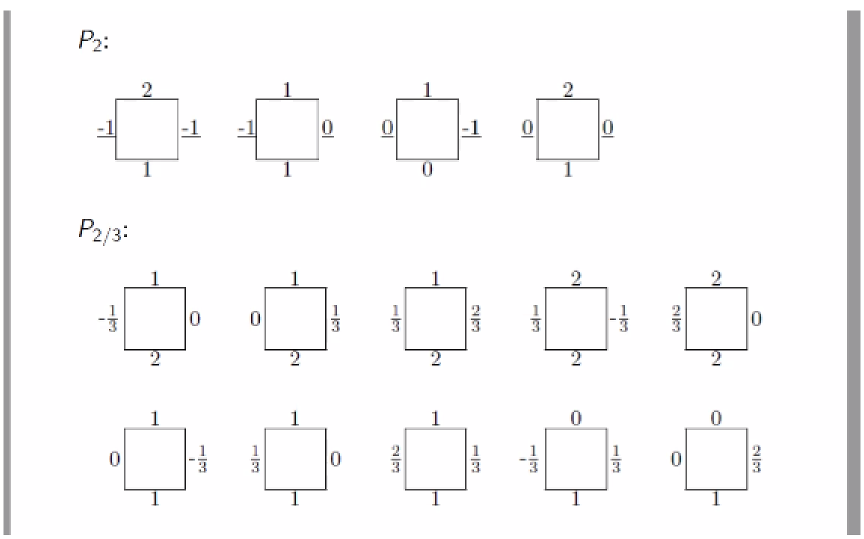


Figure 3.7: Плитки Кари

Теорема 3.1.2 (про 14 плиток). Плитками с картинки 3.7 выше можно замостить плоскость, но только апериодическим способом[a].

[a]Подчеркнутые и не подчеркнутые цифры – разные.

□ Надо доказать, что замощение существует, и что не существует периодического.

TODO: Дописать теорему ■