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

Тамарин Вячеслав

April 9, 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, меняйте. 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 $\mathbb{N} = \{0, 1, 2, \ldots\}$. Такая функция называется k-**местной частичной функцией**. Если $k = 0$, то $f = \text{const}$.

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

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

  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

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

  Оператор обозначается **R**.

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

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

[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 $f_1, \dots f_k$, что все f_i либо простейшие, либо получены из предыдущих $f_1, \dots f_{i-1}$ 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:

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

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

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

Определение 4: Частично рекурсивная функция

Функция f называется **частично рекурсивной функцией** (**ЧРФ**), если существует последовательность функций $f_1, \dots f_k$, 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

$$f(\overline{x}) = \begin{cases} y & g(\overline{x},y) = 0 \wedge y \leqslant h(\overline{x}) \wedge g(\overline{x},i) \neq 0^{b}\ \forall i < y \\ h(\overline{x})+1 & else \end{cases}$$

[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:

$$f(\overline{x}) = \sum_{y=0}^{h(x)} \prod_{i=0}^{y} \mathrm{sg}(g(\overline{x},i)).$$

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 нульмерного 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:

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

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

□

- Запишем $\pi(x,y) = \binom{x+y+1}{2} + y$. Заметим, что для $n > m$ верно
$$\binom{n}{2} - \binom{m}{2} \geqslant \binom{n}{2} - \binom{n-1}{2} = n - 1.$$
  Предположим, что $x + y > x' + y'$ и $\pi(x,y) = \pi(x',y')$. Тогда
$$y' - y = \binom{x+y+1}{2} - \binom{x'+y'+1}{2} \geqslant x + y > x' + y'.$$

Но $y \geqslant 0$, $x' \geqslant 0$, поэтому $y' - y \leqslant y$, а $x' + y' \geqslant y'$, а тогда $y' - y \leqslant x' + y'$, что противоречит полученному выше неравенству.

Если $x + y = x' + y'$, то $0 = \pi(x, y) - \pi(x', y') = y - y'$. Тогда $y = y' = 0$, поэтому $x = x'$.

Доказали, что из равенства $\pi(x, y) = \pi(x', y')$ следует равенство $(x, y) = (x', y')$.

- 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

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

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

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

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 $f(\overline{x}, i)$, $0 \leqslant i \leqslant y$.

[1] отрицательный корень можем сразу отбросить

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:

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

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_S для S.
  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 для место.
  МТ запускаем псведопараллельно (по очереди даем поработать).
  В каждой часть 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 $1^y01^{x_1}0\dots01^{x_m}001^{g(x_1,\dots x_m)}$. Для этого просто запустим M_g на входе, но не будем стирать его, а результат просто припишем после двух нулей справа.
- 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 $1^y01^{x_1}0\dots01^{x_m}01^u0^z$ 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

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

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_μ 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 — 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

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

где $\tilde{\alpha}$ — число, соответствующее α в m-ичной записи, $\tilde{q_i}$ — просто номер состояния, $\widetilde{a_j}$ — номер в алфавите (j), $\tilde{\beta}$ — 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. $\Gamma = \{a_0, a_1\}$, тогда следующее состояние будет записано как $(22, 3, 1, 13)$:

$$\underbrace{a_1 a_0 a_1 a_1 a_0}_{\alpha} q_3 \underbrace{a_1 a_1 a_0 a_1 a_1}_{\beta}$$

Кодирование команд Пусть есть переход $(q, a) \to (p, b, d)$. Сопоставим p, b, d тройку функций $\varphi_q, \varphi_a, \varphi_d$:

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

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

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

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:

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

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

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

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

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 z встречаем впервые. То есть нам нужно вычислить функцию, которая переводит

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

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

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

Тогда результатом работы МТ будет

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

■

Следствие 1. Любую частично рекурсивную функцию можно представить так, что 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], 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 $\alpha_n(x)$, которая определяется следующим образом:

$$\begin{cases} a_0(x) & = x + 1 \\ a_{n+1}(x) & = a_n^{[x+2]}(x) = \underbrace{a_n(a_n(\ldots(x)))}_{x+2 \text{ раза}} \end{cases}$$

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

□ «Доказательство – упражнение», занимает пару страниц, в ближайшее время появится здесь. ■

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 из 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 определения эквивалентны:

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

□

$\boxed{0 \Longrightarrow 2}$ 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, множеств X будет областью определения χ_X, а она вычислима.

$\boxed{1 \Longrightarrow 0}$ Пусть область определения вычисляется алгоритмом B.

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 $c(x_1, \ldots, x_m)$, 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 $\mathcal{F}^2$.

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

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

g вычислимая, U универсальная, поэтому

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

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

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

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

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

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

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:

- 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 перовое 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 неразделимые 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

[2] останавливается ли МТ M на входе x

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

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

[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)**.** Пусть $U \in \mathcal{F}^2$ — главная универсальная функция для F 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

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

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

[3]Здесь можно было бы и расписать подробно.

$\boxed{2 \Longrightarrow 1}$

□ Пусть есть такая функция $f(p,\ q)$, что $U_p \circ U_q = U_{f(p,q)}$. Хотим доказать, что U главная.

- Найдем транслятор для канторовской нумерации $t \in \mathcal{F}\colon \forall n, x \in \mathbb{N} \quad U(t(n), x) = c(n, x)$. *TODO*: дописать
- 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:

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

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 для класса $\mathcal{F}^n$, но неглавная функция $V \in \mathcal{F}^{n+1}$.

□ Пусть $U(n, x)$ — произвольная главная универсальная функция и D — множества номеров функций в U с непустой областью определения.

Заметим, что D перечислимое, так как можно построить следующий алгоритм: на шаге k будем для $i \in \{0, 1, \dots, k-1\}$ и $\bar{j} \in \{0, 1, \dots, k-1\}^n$ считать $U_i(\bar{j})$. Для этого даем на каждой из k^{n+1} машин Тьюринга поработать n 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 n, которое больше номера функции и координат какой-то точки из области определения.

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, единственный номер нигде не определенного сечения — только 0, это множество конечно, следовательно разрешимо, поэтому V неглавная по прошлому следствию. ■

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 3. 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 3.
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

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

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:

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

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 мнжеств), если любое перечислимое B m-сводится к A.

Лекция 7: †

25 march

Теорема 2.5.1. Частичная функция $f \in F_\alpha^{(1)}$, где $\alpha\colon \mathbb{N} \to \mathbb{N}$ – всюду определенная функция тогда и только тогда, когда существует перечислимое корректное множество троек M, такое что $f = M[\alpha]$.

$\boxed{1 \Longrightarrow 2}$

$\boxed{2 \Longrightarrow 1}$