

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

Тамарин Вячеслав

March 16, 2021

Contents

| | | |
|----------|--|-----------|
| 1 | Вычислимость. Система вычислимости по Клини | 3 |
| 1.1 | Рекурсивные функции | 3 |
| 1.1.1 | Простейшие функции | 3 |
| 1.1.2 | Операторы | 3 |
| 1.1.3 | Функции | 4 |
| 1.1.4 | Оператор ограниченной минимизации | 6 |
| 1.1.5 | Предикаты | 7 |
| 1.1.6 | Теоремы про рекурсии | 8 |
| 1.2 | Равносильность МТ и ЧРФ | 11 |
| 1.2.1 | Функция Аккермана | 14 |
| 2 | Разрешимые и перечислимые множества | 15 |
| 2.1 | Определения | 15 |
| 2.2 | Перечислимые множества | 15 |
| 2.3 | Универсальные функции | 18 |
| 2.3.1 | Перечислимое неразрешимое множество | 20 |
| 2.3.2 | Главные универсальные функции | 21 |
| 2.3.3 | Теорема Райса | 23 |
| 2.4 | Теорема о неподвижной точке | 24 |
| 2.5 | m -сводимость | 25 |
| 2.6 | Проблема соответствия Поста (PCP) | 27 |
| 2.7 | T -сводимость (по Тьюрингу) | 28 |

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

k -местная частичная функция, 3

кусочное задание функции, 10

общерекурсивная функция, 4

оператор минимизации, 4

оператор ограниченной минимизации, 6

оператор примитивной рекурсии, 3

оператор суперпозиции, 3

перечислимое множество, 16

предикаты, 7

примитивно рекурсивная функция, 4

проекция, 18

простейшие функции, 3

равносильность МТ и **ЧРФ**, 11

разрешимое множество, 15

рекурсия возвратная, 9

рекурсия совместная, 10

универсальная функция, 19

функция Аккермана, 14

частично рекурсивная функция, 4

Chapter 1

Вычислимость. Система вычислимости по Клини

1.1 Рекурсивные функции

Лекция 1: †

11 feb

Определение 1

Пусть функция $f: \mathbb{N}^k \rightarrow \mathbb{N}$, $k \in \mathbb{N}$, где $\mathbb{N} = \{0, 1, 2, \dots\}$. 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) — $I_n^m(x_1, \dots, x_n) = x_m$

1.1.2 Операторы

Определим три оператора:

Определение 2

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

$$h(y_1, \dots, y_m), g_i(x_1, \dots, x_n); 1 \leq i \leq n,$$

если

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

Оператор обозначается **S**.

- Функция $f^{(n+1)}_a$ получается оператором примитивной рекурсии 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 :

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

^aЗдесь и далее $f^{(n)}$ обозначается функция, принимающая n аргументов, то есть n -местная

^bподразумевается, что функция определена в этих точках

^cне определена

Пример 1.1.1.

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

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**:

$$I_1^1, s, I_3^3, h = S(s, I_3^3), f.$$

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. $\text{const}^{(n)}$

2. $x + y$

3. $x \cdot y$

4. x^y , где 0^0 можем определить, как хотим

5. $\text{sg}(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0 \end{cases}$

6. $\overline{\text{sg}}(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases}$

7. $x \div 1 = \begin{cases} u & x = 0 \\ x - 1 & x > 0 \end{cases}$

8. $x \dot{-} y = \begin{cases} 0 & x < y \\ x - y & \text{else} \end{cases}$

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. $\text{sg}(x) = \begin{cases} 0 & x = 0 \\ 1 & x \neq 0 \end{cases}$

$$\begin{cases} \text{sg}(0) & = 0 \\ \text{sg}(x + 1) & = 1 = s(0) \end{cases}$$

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

$$\mu y \leq h(\bar{x}) [g(\bar{x}, y) = 0]^a.$$

Это означает, что

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

^aЗдесь и далее $\bar{x} = x_1, \dots, x_n$.

^bАналогично, 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:

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

Внутреннее произведение равно единице только тогда, когда все $g(\bar{x}, i) \neq 0$. Если для некоторого y обнуляется $g(\bar{x}, y)$, то все произведения, начиная с $y+1$, будут равны нулю, поэтому просуммируем только y единиц. Если же такого y нет, получим сумму из $h(\bar{x}) + 1$ единиц. Именно это и нужно.

Проверим, что можно получить

$$a(\bar{x}, y) = \sum_{i=0}^y g(\bar{x}, i), \quad m(\bar{x}, y) = \prod_{i=0}^y i = 0^y g(\bar{x}, i)$$

с помощью примитивной рекурсии:

$$\begin{cases} a(\bar{x}, 0) &= g(\bar{x}, 0) \\ a(\bar{x}, y+1) &= a(\bar{x}, y) + g(\bar{x}, y+1) \end{cases} \quad \begin{cases} m(\bar{x}, 0) &= g(\bar{x}, 0) \\ m(\bar{x}, y+1) &= m(\bar{x}, y) \cdot g(\bar{x}, y+1) \end{cases}$$

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).

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

Утверждение.

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

□

- Проверим, что характеристические функции примитивно / общерекурсивны:

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

- Аналогично выразим, через простейшие:

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

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. $\text{Div}(x, y) = \begin{cases} 1, & y \mid x \\ 0, & \text{else} \end{cases}$
3. $\text{Prime}(x) = \begin{cases} 1, & x \in \mathbb{P} \\ 0, & \text{else} \end{cases}$
4. $f(x) = p_x$, где p_x — x -тое простое число, $p_0 := 2$
5. $\text{ex}(i, x)$ — степень простого числа p_i разложения x , $\text{ex}(i, 0) := 0$

□

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

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

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

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

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

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

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

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 $\pi: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$:

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

- 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

$$\binom{n}{2} - \binom{m}{2} \geq \binom{n}{2} - \binom{n-1}{2} = n - 1.$$

Предположим, что $x + y > x' + y'$ и $\pi(x, y) = \pi(x', y')$. Тогда

$$y' - y = \binom{x+y+1}{2} - \binom{x'+y'+1}{2} \geq x+y > x'+y'.$$

Но $y \geq 0$, $x' \geq 0$, поэтому $y' - y \leq y$, а $x' + y' \geq y'$, а тогда $y' - y \leq x' + y'$, 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:

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

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

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

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

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

где $\forall 1 \leq i \leq s \ t_i(y) \leq y$, $g^{(n)}, h^{(n+1+s)}, t_i^{(1)}$.

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 $m(\bar{x}, y)$, которая возвращает закодированную последовательность $f(\bar{x}, i)$, $0 \leq i \leq y$.

¹отрицательный корень можем сразу отбросить

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

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

$$f(\bar{x}, y) = \text{ith}(y, m(\bar{x}, y)).$$

Вернемся к построению m :

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



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

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

Если $g_i^{(n)}, h_i^{(k+2)}$, $1 \leq i \leq k$ — 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, \dots, c_k .

Давайте просто объединим все f_i в одну функцию

$$m(\bar{x}, y) = c(f_1(\bar{x}, y), \dots, f_k(\bar{x}, y)).$$

Теперь каждую f_i можно вычислить

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

Чтобы получить m достаточно использовать примитивную рекурсию:

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



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

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

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

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

^aНабор предикатов

^bТо есть для $i \neq j$ верно $R_i \cap R_j = \emptyset$.

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

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

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).

□

$2 \Rightarrow 1$ Если $f(x_1, \dots, x_n) = y$, то считаем, что МТ получаем $1^{x_1}01^{x_2}0 \dots 01^{x_n}$ и должна выдать 1^y ; если f не определена, МТ должна заикливиться 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:

$$(1^{x_1}01^{x_2} \dots 01^{x_n} *)^n.$$

- 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 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: †

R: Пусть рекурсия задает $f^{(m+1)}(x_1, \dots, x_m, y)$ из $g^{(m)}$ и $h^{(m+2)}$.

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

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 $1^{x_1}0 \dots 01^{x_m}01^y$ построим $1^y01^{x_1}0 \dots 01^{x_m}001^{g(x_1, \dots, x_m)}$. 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 $1^y01^{x_1}0 \dots 01^{x_m}01^u01^z$ построим $1^y01^{x_1}0 \dots 01^{x_m}01^{u+1}01^{h(x_1, \dots, x_m, u, z)}$. 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 $1^y01^{x_1}0 \dots 01^{x_m}01^u0^z$ оставим только 1^z .
- Φ : для входа $1^y01^{x_1}0 \dots 01^{x_m}01^u01^z$ 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 .

М: Хотим по МТ M_g построить M_f , вычисляющую

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

Аналогично построим несколько вспомогательных МТ:

- N_1 : приписывает 0 ко входу:

$$1^{x_1}0 \dots 01^{x_m} \longrightarrow 1^{x_1}0 \dots 01^{x_m}0.$$

- N_2 : дублирует вход, разделяя решеткой: $w \longrightarrow w\#w$
- N_3 : в продублированном входе меняет вторую половину на результат M_g

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

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

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

- N_5 : стирает все, кроме ответа

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

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

Теперь можем построить M_μ так:

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

1 \implies 2 Теперь мы хотим промоделировать работу МТ с помощью частично рекурсивной функции. На вход должны либо выдать результат, либо заиклиться. 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

$$\tilde{\alpha}, \tilde{q}_i, \tilde{a}_j, \tilde{\beta},$$

где $\tilde{\alpha}$ — число, соответствующее α в m -ичной записи, \tilde{q}_i — просто номер состояния, \tilde{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 $\tilde{Q} = \{0, 1, \dots, |Q| - 1\}$.

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 $(q, a) \rightarrow (p, b, d)$. Сопоставим p, b, d тройку функций $\varphi_q, \varphi_a, \varphi_d$:

$$\varphi_a: \tilde{Q} \times \tilde{\Gamma} \rightarrow \tilde{\Gamma}$$

$$\varphi_q: \tilde{Q} \times \tilde{\Gamma} \rightarrow \tilde{Q}$$

$$\varphi_d: \tilde{Q} \times \tilde{\Gamma} \rightarrow \{1, 2\}$$

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:

$$K = \alpha q_i a_j \beta \rightarrow \alpha' q'_i a'_j b' = K'.$$

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

Пусть

$$\begin{aligned} \tilde{q}'_i(\tilde{\alpha}, \tilde{q}_i, \tilde{a}_j, \tilde{\beta}) &= \varphi_q(\tilde{q}_i, \tilde{a}_j) \\ \tilde{\alpha}'(\tilde{\alpha}, \tilde{q}_i, \tilde{a}_j, \tilde{\beta}) &= \begin{cases} \tilde{\alpha} \cdot m + \varphi_a(\tilde{q}_i, \tilde{a}_j), & \varphi_d(\tilde{q}_i, \tilde{a}_j) = 1 \\ \left\lfloor \frac{\tilde{\alpha}}{m} \right\rfloor, & \varphi_d(\tilde{q}_i, \tilde{a}_j) = 2 \end{cases} \\ \tilde{\beta}'(\tilde{\alpha}, \tilde{q}_i, \tilde{a}_j, \tilde{\beta}) &= \begin{cases} \tilde{\beta} \cdot m + \varphi_a(\tilde{q}_i, \tilde{a}_j), & \varphi_d(\tilde{q}_i, \tilde{a}_j) = 1 \\ \left\lfloor \frac{\tilde{\beta}}{m} \right\rfloor, & \varphi_d(\tilde{q}_i, \tilde{a}_j) = 2 \end{cases} \\ \tilde{a}'_j &= \begin{cases} \tilde{\beta} \bmod m, & \varphi_d(\tilde{q}_i, \tilde{a}_j) = 1 \\ \tilde{\alpha} \bmod m, & \varphi_d(\tilde{q}_i, \tilde{a}_j) = 2 \end{cases} \end{aligned}$$

Заметим, что все эти формулы примитивно рекурсивные².

Общая работа МТ Пусть $K(0) = (\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0)$ — начальная конфигурация. Чтобы получить новую конфигурацию для шага t , посчитаем все четыре параметра:

$$\begin{aligned} K(t) = (& \\ & K_\alpha(\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0, t) \\ & K_q(\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0, t) \\ & K_a(\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0, t) \\ & K_\beta(\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0, t) \\ &) \end{aligned}$$

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

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

$$\begin{cases} K_\alpha(\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0, 0) &= \tilde{\alpha}_0 \\ K_\alpha(\tilde{\alpha}_0, \tilde{q}_0, \tilde{a}_0, \tilde{\beta}_0, t+1) &= \tilde{\alpha}'(K_\alpha(\dots, t), K_q(\dots, t), K_a(\dots, t), K_\beta(\dots, t)) \end{cases}$$

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

$$\tilde{a}_0 + \tilde{b}_0 m \longrightarrow a_z + \beta_z m,$$

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

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

Тогда результатом работы МТ будет

$$\begin{aligned} \varphi(x) = & m \cdot K_\beta\left(0, 0, x \bmod m, \left\lfloor \frac{x}{m} \right\rfloor, \right. \\ & \left. \mu t [K_q(0, 0, x \bmod m, \left\lfloor \frac{x}{m} \right\rfloor, t) = z] + \right. \\ & \left. + K_a(0, 0, x \bmod m, \left\lfloor \frac{x}{m} \right\rfloor) \right) \end{aligned}$$

■

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, вычисляемая за примитивно рекурсивное время ^a, является примитивно рекурсивной.

^aвремя, 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:

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

Теорема 1.2.2. $\alpha_n(n): \mathbb{N} \rightarrow \mathbb{N}$ 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.

^aЭто 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).

□

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.

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 i ;
- 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: †

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

□

0 \implies 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.

3 \implies 1 Действительно, множество X будет областью определения χ_X , а она вычислима.

1 \implies 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.

2 \implies 0 Аналогично, но выводим значение функции на элементе, на котором мы останавливаемся.

1 \implies 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.

0 \implies 4 Пусть A — алгоритм, перечисляющий X . Рассмотрим любой $n_0 \in X$.

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

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

4 \implies 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 \bar{A} перечислимые.

□

1 \implies 2 Так как A разрешимо, можем рассмотреть вычислимую характеристическую функцию $\chi_A(n)$.

Построим полухарактеристические для A и \bar{A} :

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

2 \implies 1 Пусть мы хотим проверить $n \stackrel{?}{\in} A$.

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

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

Определение 11: Проекция

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

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

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

□

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

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

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

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

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

□

$1 \implies 2$ 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 $(x, f(x))$.

$2 \implies 1$ 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 |
| полурешимое | |
| вычислимо перечислимое | вычислимое |
| полурекурсивное | рекурсивное |
| semi-decidable | decidable |
| semi-recursive | recursive |
| enumerable | |

2.3 Универсальные функции

Определение 12

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

Определение 13: Универсальная функция

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

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

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

$$U'(n, \bar{x}) := U(n, c(\bar{x})),$$

где U — универсальная для \mathcal{F}^2 .

Во-первых, заметим, что все сечения вычислимы.

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

Определим

$$g(y) := f(c_1(y), \dots, c_m(y)).$$

g вычислима, U универсальная, поэтому

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

$$\begin{aligned}
U^l(n, \bar{x}) &= && \text{(по определению } U) \\
= U(n, c(\bar{x})) &= && (n - \text{номер } g) \\
= g(c(\bar{x})) &= && \text{(по определению } g) \\
= f(c_1(c(\bar{x})), \dots, c_m(c(\bar{x}))) &= f(\bar{x})
\end{aligned}$$

То есть U^l 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:

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

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

С другой стороны, $d^l(n)$ отличается от всех сечений U : если $\forall x: U(n, x) = d^l(x)$, 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.

^aто есть нельзя доопределить до всюду определенной вычислимой

□ Подходит функция $d^l(n) = U(n, n) + 1$, 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 $W_n = \{\bar{x} \mid (n, \bar{x}) \in W\}$.

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 $X \in \mathbb{N}^m$.

¹далее 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. \bar{F} — пример непечислимого множества.

Замечание. Область определения универсальной функции перечислимо, 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 $f: \mathbb{N} \rightarrow \{0, 1\}$, 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 непересекающиеся неразделимые множества.

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

Тогда χ_S — общерекурсивное дополнение d''' . Противоречие. ■

Лекция 4: †

2.3.2 Главные универсальные функции

²останавливается 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

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

^aпо 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 $U(x, \bar{y}) := T(l(x), r(x), \bar{y})$.

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

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

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

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

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

□

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

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

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

³Здесь можно было бы и расписать подробно.

2 \Rightarrow 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 $t \in \mathcal{F}$: $\forall n, x \in \mathbb{N} \quad U(t(n), x) = c(n, x)$.
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 :

$$\begin{aligned} U_a(U(t(n), x)) &= U_a \circ U_{t(n)}(x) = V_n(x) \\ \parallel \\ U_{f(a, t(n))}(x) &= V_n(x) \end{aligned}$$

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: $\emptyset \subsetneq \mathcal{A} \subsetneq \mathcal{C}$.

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:

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

Заметим, что V вычислима, 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

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

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 $\forall n: n \not\equiv h(n)$.

^aТо 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 $\forall n: h(n) \not\equiv n$.

Рассмотрим $t(x) := h(g(x))$ 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 $\exists n: U_n = U_{h(n)}$.

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

Покажем, что выполняется первый пункт из [леммы 3](#).

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

Так как U главная существует транслятор:

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

Проверим, что s и есть \equiv -продолжение f

- если $f(n)$ определена, то $U_{s(n)} = U_{f(n)}$, то есть $s(n) \equiv f(n)$.
- 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

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

□ 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 $s(n)$:

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

■

2.5 m -сводимость

Определение 18: m -сводимость

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

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

Свойства.

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

□

- Чтобы проверить $x \stackrel{?}{\in} A$, 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 $f = \text{id}$, 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 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 :

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

Проверим, что это множество подходит.

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

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

Тогда

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

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

Замечание. Если K — m -полное, $K \leq_m A$ — 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.

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

Так как U главная

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

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 (РСР)

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 $h_1, h_2: \Delta^* \rightarrow A^*$. 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 строка 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 вычисления МТ.

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

1. Начальные доминошки $(\varepsilon, \triangleright q_0 \#)$, $x \in \Sigma \cup _$
2. Доминошки копирования (c, c) , $c \in A$
3. Для всех правил $(q, c) \rightarrow (q', c', +1): (qc, c'q')$, если $c = _$, добавляем еще доминошку $(q\#, c'q'\#)$.
4. Для всех правил $(q, c) \rightarrow (q', c', -1): (aqc, q'ac')$, 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 копирования

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 РСР разрешима следующим образом:

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

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

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

■

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

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

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

2.7 Т-сводимость (по Тьюрингу)

Определение 20: Сводимость по Тьюрингу

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

Свойства.

- $B \leq_m A \implies B \leq_T A$
- $\forall A: A \leq_T \mathbb{N} \setminus A$
- сводимость по тьюрингу транзитивна и рефлексивна
- $A \leq_T B$ и B разрешимо, то A тоже разрешимо

□

■

Замечание. Если A перечислимо, $B \leq_T A$, 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 может обратиться к оракулу — попросить оракула вычислить $f(n)$.