Дискретная математика 2. Конспект

Бобень Вячеслав
@darkkeks, GitHub

2020

> “Здесь должна быть чья-то цитата”.
>
> — Bottom text

Содержание

1 Лекция 1. **2**

1 Лекция 1.

Мы будем рассматривать алгоритм интуитивно, не давая ему точного определения. Вместо определения алгоритма мы будем постепенно неформально описывать его свойства.

1. Алгоритмов счётно много. (их можно отождествить с словами конечного алфавита)
2. Алгоритм исполняется по шагам.
3. Алгоритм работает конечно много шагов или зацикливается.

   Например, `while(1);` (вычисляет нигде не определенную функцию)
4. Алгоритм принимает вход и может подавать что-то на выход.

   Сделать алгоритм, принимающий на вход число $x \in \mathbb{R}$ и возвращающий $\sin x$ мы не можем. Хотя некоторые числа имеют конечное описание, в общем случае вещественное число это бесконечный поток байт. То есть в общем случае обработать бесконечный поток байт конечной процедурой мы не можем.

   Считаем, что вход и выход — тоже слова конечного алфавита.

   Так как слов конечного алфавита счётно много, нам удобно отождествить их с натуральными числами. (или $\mathbb{N}^2$, $\mathbb{N}^3$, ...)

Алгоритм может вычислять функцию $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$. Здесь буква p означает, что алгоритм вычисляет *частичную функцию*.

Определение 1.1. Алгоритм $\mathcal{F}$ вычисляет функцию $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$, если $\forall x \in \mathbb{N}$

$$x \in \operatorname{dom} f \implies \text{алгоритм } \mathcal{F} \text{ на входе } x \text{ останавливается и выводит } f(x),$$
$$x \notin \operatorname{dom} f \implies \text{алгоритм } \mathcal{F} \text{ на входе } x \text{ НЕ останавливается на за какое конечное число шагов.}$$

Определение 1.2. Функция f *вычислима*, если существует алгоритм, который её вычисляет.

Пример.

$$f(x) = \begin{cases} 1, & \text{если бог есть,} \\ 0, & \text{иначе.} \end{cases}$$

Утверждается, что функция f вычислима, так как существует алгоритм, ее вычисляющий. (либо всегда возвращающий 1, либо 0). Например,

```
int f(int x) {
    return 1;
}
```

Определение 1.3. Множество $A \subseteq \mathbb{N}$ *разрешимо*, если $\exists$ алгоритм $\mathcal{A}$ такой, что $\forall x \in \mathbb{N}$

$$x \in \mathcal{A} \implies \text{алгоритм выводит 1 и останавливается}$$
$$x \in \mathcal{A} \implies \text{алгоритм выводит 0 и останавливается.}$$

Утверждение 1.1. *A разрешимо $\iff$ вычислима характеристическая функция $\chi_A\colon \mathbb{N} \to \{0,1\}$ множества A:*

$$\chi_A(n) = \begin{cases} 1, & n \in A, \\ 0, & n \notin A. \end{cases}$$

Утверждение 1.2. *Существует неразрешимое множество.*

Доказательство. Алгоритмов лишь счётно много. Подмножеств $\mathbb{N}$ несчётно много. ■

Следствие 1.1. $\exists$ невычислимая функция.

Утверждение 1.3. *Если A конечно, то A разрешимо.*

Доказательство. $A = \{a_1, \ldots, a_n\}$. Тогда, характеристическая функция A:

```
int in_A(x) {
    return (x == a1) || (x == a2) || ... || (x == a_n);
}
```

■

Утверждение 1.4. *Если A, B разрешимы, то разрешимы $A \cup B$, $A \cap B$, $\overline{A}$, $A \times B$.*

Доказательство.

- $\chi_{A\cap B}(n) = \chi_A(n) \cdot \chi_B(n)$;
- $\chi_{A\cup B}(n) = \chi_A(n) + \chi_B(n)$;
- $\chi_{\overline{A}} = 1 - \chi_A(n)$;
- $\chi_{A\times B}(n, m) = \chi_A(n) \cdot \chi_B(m)$. ■

Определение 1.4. Множество $A \subseteq \mathbb{N}$ называется *перечислимым*, если существует алгоритм $\mathcal{A}$ («перечислитель»), такой что работая на пустом входе (или любом) алгоритм $\mathcal{A}$ никогда не останавливается, но в процессе работы выводит все элементы множества A и только их.

То есть для любого элемента $x \in A$ существует конечный момент времени, в который алгоритм $\mathcal{A}$ выведет элемент x.

Утверждение 1.5. *Если A разрешимо, то A перечислимо. (обратное неверно)*

Доказательство.

```
n = 0;
while (true) {
    if (in_A(n)) {
        return n;
    }
    ++n;
}
```

■

Утверждение 1.6. *Из разрешимости не следует конечность. Например, $\mathbb{N}$ или $2\mathbb{N}$ (множество четных чисел).*

Теорема 1.1 (Поста)**.** *A разрешимо $\iff$ A и $\overline{A}$ перечислимы.*

Доказательство.

$\Longrightarrow$ A разрешимо $\Longrightarrow$ A перечислимо.
A разрешимо $\Longrightarrow$ $\overline{A}$ разрешимо $\Longrightarrow$ $\overline{A}$ перечислимо.

$\Longleftarrow$ Дано n, хотим посчитать $\chi_A(n)$.
Пусть $\mathcal{A}$ перечисляет A и $\mathcal{B}$ перечисляет $\overline{A}$. Поочередно будем делать по шагу алгоритмов $\mathcal{A}$ и $\mathcal{A}$.
Если на каком-то шаге $\mathcal{A}$ вывел n, то $n \in A$. Выводим 1 и останавливаемся.
Если на каком-то шаге $\mathcal{B}$ вывел n, то $n \in \overline{A}$. Выводим 0 и останавливаемся.
Так как $A \cup \overline{A} = \mathbb{N}$, то одно из событий точно случится, следовательно наш алгоритм обязательно завершится. ■

Определение 1.5. Проекция множества $A \subseteq \mathbb{N}^k$:

$$\mathrm{pr}^i A = \{b \in \mathbb{N} \mid (a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n) \in A\}.$$

Утверждение 1.7. *Если A и B перечислимы, то также перечислимы множества $A \cup B$, $A \cap B$, $A \times B$, $\mathrm{pr}^i A$.*

Доказательство. Пусть $\mathcal{A}$ перечисляет A и $\mathcal{B}$ перечисляет B.

$\mathrm{pr}^i A$ Запустим $\mathcal{A}$ и для каждого выведенного набора $(a_1, \ldots, a_n)$ будем печатать i-ю его координату.

$A \cup B$ Будем поочередно делать шаги перечислителей $\mathcal{A}$ и $\mathcal{B}$. Все выведенные ими числа отправляем в выходной поток.

$A \cap B$ Будем поочередно делать шаги перечислителей $\mathcal{A}$ и $\mathcal{B}$.
Будем добавлять весь вывод $\mathcal{A}$ в буффер A', аналогично для $\mathcal{B}$ и B'.
$A'_i :=$ то, что лежит в буфере A' после i-го шага $\mathcal{A}$.
Аналогично для B'_i.
После того, как мы сделали i-й шаг $\mathcal{A}$ и $\mathcal{B}$ выводим множество $A'_i \cap B'_i$ — оно конечно; это делается за конечное время.

$A \times B$ Все аналогично $A \cap B$, но выводим $A'_i \times B'_i$. ■

Определение 1.6. Пусть $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$. Тогда, *график* функции:

$$\Gamma_f = \left\{(x, y) \in \mathbb{N}^2 \mid f(x) = y\right\}.$$

Теорема 1.2 (о графике). *Пусть $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$. Тогда f вычислима $\iff$ Γ_f перечислим.*

Доказательство.

$\Longleftarrow$ Вход: x. Хотим $f(x)$ (если $\exists$).

Алгоритм $\mathcal{F}$: Запускаем перечислитель Γ_f и ждем первой пары вида (x, y).

$(x, y) \in \Gamma_f \implies y = f(x)$.

Теперь выводим y и завершаемся.

Если же x не принадлежит $\operatorname{dom} f$, то мы будем ждать такой пары бесконечно (зациклимся).

$\Longrightarrow$ Дано: f и вычисляющий ее алгоритм $\mathcal{F}$.

Хотим сделать перечислитель для Γ_f.

Знаем, что $\mathbb{N}^k$ перечислимо как декартово произведение $\mathbb{N}$.

Запустим перечислитель для $\mathbb{N}^3$. Для каждой выведенной тройки $(x, y, k) \in \mathbb{N}^3$ делаем k шагов $\mathcal{F}$ на входе x. Если он вывел y и остановился, то $f(x) = y$. Тогда напечатаем пару (x, y).

Допустим, $(x, y) \in \Gamma_f$. Тогда $f(x) = y$. Значит, существует такое число шагов k, что $\mathcal{F}$ на входе x выведет y за k шагов. Мы рассмотрим тройку (x, y, k) и выведем (x, y). ■

Следствие 1.2. Если f вычислима и A перечислимо, то $f(A)$ перечислимо.

Доказательство. $f(A) = \operatorname{pr}^2(\Gamma_f \cap (A \times \mathbb{N}))$.
f вычислима $\implies$ Γ_f перечислим.
A перечислимо $\implies$ $A \times \mathbb{N}$ перечислимо.
Таким образом $\Gamma_f \cap (A \times \mathbb{N})$ перечислим, а значит перечислимо и $\operatorname{pr}^2(\Gamma_f \cap (A \times \mathbb{N}))$.
Аналогично $f^{-1}(A) = \operatorname{pr}^1(\Gamma_f \cap (\mathbb{N} \times A))$. ■

Следствие 1.3. Если $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$ вычислима, то $\operatorname{dom} f$ и $\operatorname{rng} f$ перечислимы.

Доказательство. $\operatorname{dom} f = f^{-1}(\mathbb{N})$, $\operatorname{rng} f = f(\mathbb{N})$. ■

Определение 1.7. Если $A \subseteq \mathbb{N}$, то *полухарактеристичекая* функция $\omega_A\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$ множества A:

$$\omega_A(n) \simeq \begin{cases} 1, & n \in A, \\ \text{не определена}, & n \notin A. \end{cases}$$

Определение 1.8. $f(x) \simeq g(x)$ («совпадает») — или $f(x)$ и $g(x)$ оба определены и равны, либо оба не определены для любого x.

Замечание. $\operatorname{dom} \omega_A = A$.

Утверждение 1.8. *Если A перечислимо, то ω_A вычислимо.*

Доказательство. Алгоритм для ω_A, вход: n.
Запускаем перечислитель A и ждем появляения n; когда появится, выводим 1 и останавливаемся. ■

Предложение 1.1. Если ω_A вычислима, то A перечислимо.

Доказательство. $A = \operatorname{dom} \omega_A$. ■

Следствие 1.4. A перечислимо $\iff$ ω_A вычислима (A полуразрешимо).

Утверждение 1.9. *Если A перечислимо и $A \neq \varnothing$, то $\exists$ вычислимая тотальная $f\colon \mathbb{N} \to \mathbb{N}$, такая что $A = \operatorname{rng} f$ (то есть $A = \{f(0), f(1), \dots\}$).*

Доказательство. У A есть перечислитель $\mathcal{A}$.
Так как $A \neq \varnothing$, то $\mathcal{A}$ напечатает какое-то число $a \in A$. Пусть впервые напечатает на шаге k.
Пусть $f(0) := a$, а $f(n+1) :=$ последнее число, которое $\mathcal{A}$ напечатает за $n + k + 1$ шагов.
Очевидно, что f вычислима. ■

Следствие 1.5. Если A перечислимо, то $\exists$ вычислимая $f\colon \mathbb{N} \xrightarrow{p} \mathbb{N}$, такая что $A = \operatorname{rng} f$.

Утверждение 1.10. *Если $A \subseteq \mathbb{N}$ перечислимо, то $\exists$ разрешимое $B \subseteq \mathbb{N}^2$, такое что $A = \operatorname{pr}^1 B$.*

Доказательство. Пусть $\mathcal{A}$ — перечислитель A.
$B := \big\{(n, k) \in \mathbb{N}^2 \mid \text{ алгоритм } \mathcal{A} \text{ выводит } n \text{ на шаге } k\big\}$.
$\operatorname{pr}^1 B = \{n \in \mathbb{N} \mid \exists k : (\text{алгоритм } \mathcal{A} \text{ выводит } n \text{ на шаге } k)\} = \{n \in \mathbb{N} \mid n \in A\} = A$. ■

Теорема 1.3 (равносильное определение перечислимого множества). $\forall A \subseteq \mathbb{N}$ *следующие утверждения равносильны:*

1. A *перечислимы;*
2. A *полуразрешимо;*
3. $\exists f$ *вычислимая* $: \mathbb{N} \xrightarrow{p} \mathbb{N} : A \operatorname{dom} f$*;*
4. $\exists f$ *вычислимая* $: \mathbb{N} \xrightarrow{p} \mathbb{N} : A \operatorname{rng} f$*;*
5. $A = \varnothing$ *или* $\exists f$ *вычислимая тотальная* $: \mathbb{N} \to \mathbb{N} : A = \operatorname{rng} f$*;*
6. $\exists$ *разрешимое* $B \subseteq \mathbb{N}^2 : A = \operatorname{pr}^1 B$.