

Рекурсивні функції

Власенко Д. І. Курінний Г.Ч. Невмержицька О.М. Шугайло О.О.

Травень — 2015

1 Обчислюваність. Рекурсивні функції

1.1 Загальні слова про обчислюваність, алгоритми та тези ¹

Мова йтиме про функції, чиї аргументи належать розширеному натуральному ряду — $\mathbb{N} \cup \{0\} = \overline{\mathbb{N}}$. Нас цікавитимуть функції, значення яких ми можемо обчислити в тому розумінні, що є алгоритм, застосувавши який до аргументів ми через певну кількість кроків одержимо значення функції. Алгоритм розуміємо інтуїтивно. Такі функції називатимемо обчислюваними. Алгоритмів (вважаємо, що їх можна записати словами) зліченна кількість, а характеристичних функцій підмножин множини $\overline{\mathbb{N}}$ стільки ж, скільки підмножин у $\overline{\mathbb{N}}$ — більше ніж зліченна кількість. Тому є не обчислювані функції.

Формальні уточнення інтуїтивного поняття “обчислюваність” та “алгоритм” запропоновані в 1936 році (Гьодель, Ербран, Чьорч, Кліні, Тьюрінг, ...) Пізніше свої уточнення запропонували Пост, Марков, Шепердсон, Стерджіс, ... Наш вибір того чи іншого підходу до обчислюваності суб’єктивний, — але це не важливо, оскільки є дослідження, які показують рівно-

¹Послідовне навчальне викладення обчислюваності можна знайти в книзі К.Катленд “Вычислимость. Введение в теорию рекурсивных функций.” М.:Мир, 1983

сильність усіх підходів. Всі підходи закінчуються тезою, яка стверджує,

Будь-який інтуїтивний, неформальний алгоритм може бути модельований в запропонованому підході.

“Моделювання” означає заміну справжніх предметів, з якими має справу інтуїтивний алгоритм, на символи, з якими має справу підхід, а найпростіші дії інтуїтивного алгоритма над предметами, заміняються на найпростіші переходи від одних символів до інших. В такому мовному оточенні часто вживається також слово “кодування”. Кодування означає перезапис інформації з метою зручності обробки, а в каналах зв’язу також для захисту від випадкового пошкодження. Тепер можна сказати уже досить складне речення — при моделюванні справжнього алгоритма початкові дані кодуються. Для прикладу, коли справжній алгоритм полягає в тому, щоб дві корови зігнати з трьома коровами в одну череду, то у відповідній машині Тьюрінга дві корови закодуються трьома одиницями, а три корови — чотирма одиницями. А результатом виконання машини Тьюрінга, яка моделює наш алгоритм, буде шість одиниць на стрічці і машина Тьюрінга знаходиться у кінцевому стані.

Звичайно згадана теза носить ім’я того математика, який запропонував підхід. Таким чином, якщо підхід (моделювання машинами Тьюрінга) запропонував Тьюрінг, то згадана теза уде звучати так

Будь-який інтуїтивний, неформальний алгоритм може бути модельований на машині Тьюрінга,

і називатися тезою Тьюрінга, Відповідно, є теза Чьорча, Поста, ...

Свідченнями на користь прийняття тези на віру, є такі

1. Різні незалежні дослідники запропонували незалежні підходи, які виявилися рівносильними — все, що можна зробити одним підходом, можна зробити іншими.
2. Для дуже багатьох інтуїтивно обчислюваних функцій доведена їх формальна обчислюваність, уже доведено, що дуже багато інтуїтивних алгоритмів м оделюються запропонованими підходами.
3. Ніхто не зумів знайти функцію, яка була б обчислювана в інтуїтивному розумінні, і яку не можна було б обчислити на машині Тьюрінга, чи за допомогою іншого (рівносильного) формального підходу, ніхто не зумів придумати інтуїтивно здійснений алгоритм, який не можна було б здійснити на тій чи іншій моделі алгоритма.

Для математика з досвідом роботи з абгоритмами є додаткове свідоцтво на кор исть прийняття тези про обчислюваність, — це його власний досвід переведення інтуїтивного алгоритма у формальний. Для математика без досвіду роботи у царині алгоритмів прийняття чи не прийняття тези залежить від того, наскільки від може довіритися досвіду інших там, де його власний досвід відсутній чи просто малий.

Підхід Тьюрінга ми уже розібрали. Переходимо до іншого підходу.

1.2 Визначення частково рекурсивних, загально рекурсивних та примітивно рекурсивних функцій. Теза Чорча.

Частково рекурсивні функції визначаються індуктивно — ці функції будуються по кроках, 1-й, 2-й та т.п.

База індукції. Вводимо три найпростіші функції, і вважаємо їх такими, що побудовані на першому кроці.

Перша функція $s(x) = x + 1$ — знаходження наступного числа або “плюс один”. Це функція від однієї змінної. $s(5) = 6$, $s(0) = 1$.

Друга функція — тотожний нуль. Це нульмісна функція, або виділений елемент в $\mathbb{N} \cup \{0\}$.

І третя функція — проектування²

$$I_m^n(x_1, x_2, \dots, x_n) = x_m, \quad 1 \leq m \leq n.$$

Ця функція від кількох змінних, вона вибирає один аргумент із списку. Тут можна вважати, що маємо нескінченно багато функцій (маємо два параметри — із скількох аргументів вибираємо, і який саме аргумент. Задавши ці два параметри одержуємо одну функцію). $I_2^5(2, 3, 0, 7, 1) = 3$.

Інтуїтивно, згадані три функції є обчислюваними. Звернемо увагу, що всі три найпростіші функції є всюди визначені, ми їх можемо підрахувати при будь-яких наченнях аргументів.

Індуктивне припущення. Припускаємо, що ми вже знаємо, які функції одержані на n -му ($n \in \mathbb{N}$) кроці.

Індуктивний перехід. На $n + 1$ -му ($n \in \mathbb{N}$) кроці одержуємо функції із тих, що побудовані на попередніх кроках за допомогою трьох операторів (правил одержання нових функцій із старих) — оператора суперпозиції, оператора примітивної рекурсії, та оператора мінімізації. Що це за оператори — поясним нижче.

Таким чином, кожна частково рекурсивна функція з’явилася на цілком певному кроці. Тут мається на увазі функція як правило обчислення. І різні правила (отже, різні функції) можуть давати при всіх значеннях змінних один і той же результат.

Перший оператор — оператор суперпозиції.

Оператор C суперпозиції можна застосувати до $n + 1$ -ї функції $f, g_i (i = 1, 2, \dots, n)$ і одержати функцію $h = C(f, g_1, g_2, \dots, g_n)$

²Її називають також функцією вибору, селективною функцією

наступним чином: якщо маємо

$$f(y_1, y_2, \dots, y_n), \quad g_i(x_1, x_2, \dots, x_m),$$

то

$$h(x_1, x_2, \dots, x_m) = f(g_1(x_1, x_2, \dots, x_m), g_2(x_1, x_2, \dots, x_m), \dots, g_n(x_1, x_2, \dots, x_m))$$

Тут обов'язково перша функція має n аргументів, за кількістю решти функцій. Отже коли $n = 2$, то перша функція $f(x_1, x_2)$ повинна мати 2 аргументи, а дві функції g_1, g_2 можуть бути від будь-якої кількості будь-яких аргументів. Щоб вони мали один і той же набір аргументів, можна вводити фіктивні змінні. Для прикладу, візьмемо

$$f(x_1, x_2) = 2x_1 + 3x_2, \quad g_1 = 2^{x_1} + y, \quad g_2 = u^3 + v^4 + 5.$$

Тоді

$$C(f, g_1, g_2) = h(x_1, y, u, v) = 2(2^{x_1} + y) + 3(u^3 + v^4 + 5).$$

Знову звернемо увагу, що коли оператор суперпозиції застосовується до всюди визначених функцій, то він дає всюди визначену функцію. В інших розділах математики, замість оператора суперпозиції говорять про композицію функцій, про складні функції, функції від функцій.

Оскільки функція 0 має нуль аргументів, то $C(0) = 0$.

В курсі математичного аналізу розглядають функцію $0(x)$ — її графіком є вісь OX , функцію $0(x, y)$ — вона задає площину OXY . В такому підході може виникнути неприємна ситуація, коли замість змінних у функцію $0(x, y)$ потрібно підставити всюди невизначені функції. В нашому підході такі ситуації заборонені — у функції 0 немає аргументів.

Переходимо до наступного оператора — оператора примітивної рекурсії Пр.

Оператор примітивної рекурсії³ (позначаємо його Пр) можна застосувати до двох функцій $\phi(x_1, x_2, \dots, x_n)$, $\psi(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2})$ і одержати функцію $f(x_1, x_2, \dots, x_n, x_{n+1})$ за допомогою двох правил. Перше правило (воно повинно нагадувати базу індуктивного визначення)

$$f((x_1, x_2, \dots, x_n, 0) = \phi(x_1, x_2, \dots, x_n)$$

і друге правило (воно повинно нагадувати індуктивний перехід)

$$f((x_1, x_2, \dots, x_n, y + 1) = \psi(x_1, x_2, \dots, x_n, y, f((x_1, x_2, \dots, x_n, y))$$

Звернемо увагу, що вище n означало кількість змінних. Ця кількість може бути нулем. Отже функція ϕ може бути нульмісною. Тоді функція ψ має бути двомісною (змінні можуть бути фіктивні⁴), і за допомогою оператора суперпозиції одержуємо 1-місну функцію — змінна може бути фіктивною.

Для прикладу, застосуємо оператор примітивної рекурсії до функцій

$$\phi(x_1) = x_1 = I_1^1(x_1), \quad \psi(x_1, x_2, x_3) = x_3 + 1 = s(x_3).$$

Функція ϕ від однієї змінної, функція ψ від трьох змінних, отже будуємо функцію $f(x_1, x_2)$ від двох змінних:

$$f(x_1, 0) = x_1, \quad f(x_1, 1) = f(x_1, 0) + 1 = x_1 + 1,$$

$$f(x_1, 2) = f(x_1, 1) + 1 = x_1 + 2, \quad f(x_1, 3) = f(x_1, 2) + 1 = x_1 + 3, \dots$$

Вгадуємо, $f(x_1, x_2) = x_1 + x_2$. Вгадане можна обґрунтувати методом матіндукції. Таким чином

$$x_1 + x_2 = \text{Пр}(I_1^1(x_1), s(I_1^3(x_1, x_2, x_3))). \quad (1)$$

³Оператор примітивною рекурсії називають також схемою примітивної рекурсії, подібно до того, як ми виписували схеми аксіом в численні висловлювань.

⁴фіктивні змінні обговорювалися в булевих функціях. Тут користуємося нестрогим роз'ясненням: змінна фіктивна, якщо від неї функція не залежить

За допомогою оператора примітивної рекурсії можна визначити функції $0(x), 0(x, y), \dots$. Так якщо взяти $\phi = 0$ — нульмісна функція, і $\psi(x_1, x_2) = x_1$, то результатом роботи оператора примітивної рекурсії буде функція $0(x)$.

Сталі функції від однієї змінної (отже ця змінна є фіктивною) визначаються правилом: $f(y+1) = f(y)$. Тому для визначення сталої функції можна брати функцією ϕ кількаразову суперпозицію функції s і 0 . А функцією $\psi(x_1, x_2)$ можна брати $\psi(x_1, x_2) = x_2$. Так, якщо застосувати оператор примітивної рекурсії до функцій

$$\phi = s(s(s(s(0))))), \quad \psi(x_1, x_2) = x_2,$$

то одержимо сталу функцію $f(x) = 4$.

Застосуємо оператор примітивної рекурсії до функцій

$$\phi(x_1) = 0(x_1), \quad \psi(x_1, x_2, x_3) = x_1 + x_2.$$

Функція ϕ від однієї змінної, функція ψ від трьох змінних, отже будуємо функцію $f(x_1, x_2)$ від двох змінних:

$$f(x_1, 0) = 0, \quad f(x_1, 1) = 0 + f(x_1, 0) = x_1,$$

$$f(x_1, 2) = x_1 + f(x_1, 1) = 2x_1, \quad f(x_1, 3) = x_1 + f(x_1, 2) = 3x_1, \dots$$

Вгадуємо, $f(x_1, x_2) = x_1 \cdot x_2$. Вгадане можна обґрунтувати методом математичної індукції. Таким чином

$$x_1 \cdot x_2 = \text{Пр}(0, x_1 + x_2). \quad (2)$$

Прямою перевіркою переконуємося, що застосувавши оператора примітивної рекурсії до функцій $\phi(x_1) = 0 = 0(x_1)$, $\psi(x_1, x_2, x_3) = s(0)$, одержимо функцію

$$f_1(x) = \begin{cases} 0, & \text{якщо } x = 0, \\ 1, & \text{якщо } x \neq 0, \end{cases} \quad (3)$$

Прямою перевіркою переконуємося, що застосувавши операторо примітивної рекурсії до функцій $\phi(x_1) = 1 = s(0(x_1))$, $\psi(x_1, x_2, x_3) = 0$, одержимо функцію

$$f_2(x) = \begin{cases} 1, & \text{якщо } x = 0, \\ 0, & \text{якщо } x \neq 0, \end{cases} \quad (4)$$

Застосувавши оператор примітивної рекурсії до функції $\phi(x_1) = 1 = s(0(x_1))$, та функції $\psi(x_1, x_2, x_3) = f_2(x_2)$, що визначена формулою (4) одержимо функцію

$$f_3(x) = \begin{cases} 1, & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ 0, & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases} \quad (5)$$

Застосувавши оператор примітивної рекурсії до функції $\phi(x_1) = 1 = 0$, та функції $\psi(x_1, x_2, x_3) = f_2(x_2)$, що визначена формулою (4) одержимо функцію

$$f_4(x) = \begin{cases} 0, & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ 1, & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases} \quad (6)$$

Оператор мінімізації застосовують до однієї функції $g(x_1, x_2, \dots, x_n, x_{n+1})$ і одержують функцію $f(x_1, x_2, \dots, x_n, x_{n+1})$. Значення y цієї функції $y = f(x_1, x_2, \dots, x_n, x_{n+1})$ визначається як розв'язок рівняння

$$g(x_1, x_2, \dots, x_n, y) = x_{n+1}$$

Серед усіх розв'язків шукаємо найменший. Розв'язок шукаємо підставляючи по черзі замість y числа $0, 1, 2, \dots$. Перший знайдений розв'язок і буде потрібним. Застосування цього оператора повинно нагадувати знаходження оберненої функції.

Для прикладу візьмемо $n = 1$ і застосуємо оператор мінімізації до $g(x) = x + 1$.

Знаходимо значення $y = f(0)$ $g(y) = 0$, тобто $x + 1 = 0$. Оскільки розв'язку це рівняння не має, то значення $f(0)$ немає, функція f не визначена в точці 0. А для $x \geq 1$ $f(x) =$

$x - 1$. Застосувавши оператор примітивної рекурсії до функції $\phi(x_1) = 0$, та функції $\psi(x_1, x_2, x_3) = x - 1$,

$$x \dot{-} 1 = \begin{cases} 0, & \text{якщо } x = 0, \\ x - 1, & \text{якщо } x \neq 0. \end{cases} \quad (7)$$

Якщо $g(x) = 2x$, то $f(x) = x/2$ для парних x і $f(x)$ не визначена для непарних.

Нехай функції f_3, f_4 визначені формулами (5)(6). Тоді

$$f_{ev}(x) = f_3(x) \cdot x = \begin{cases} x, & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ 0, & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases}$$

і

$$f_{odd}(x) = f_4(x) \cdot x = \begin{cases} 0, & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ x, & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases}$$

Із цих функцій та функції $x_1 + x_2$ за допомогою оператора суперпозиції одержуємо функцію

$$\left[\frac{x}{2} \right] = \frac{f_{ev}(x)}{2} + \frac{f_{odd}(x) \dot{-} 1}{2}. \quad (8)$$

Розглянемо сталу функцію $g(x) = 5$. Щоб знайти значення $y = f(0)$ ми повинні розв'язати рівняння $g(x) = 0$, тобто $5=0$. Це рівняння не має розв'язку, отже $f(0)$ не визначено. Також не визначено $f(1), f(2), f(3), f(4), f(x)$ коли $x > 5$. Знайдемо $f(5)$. Для цього розв'язуємо рівняння $5 = 5$. Ми маємо вельми точну вказівку, як саме шукати розв'язок — потрібно по черзі підставляти замість змінної числа $0, 1, 2, \dots$. Оскільки $5=5$ завжди, то $5=5$ і при першій же перевірці — коли змінна дорівнює 0 . Отже $y = 0$ і $f(5) = 0$.

Приклади показують, що оператор мінімізації суттєво відрізняється від перших двох — суперпозиції та примітивної рекурсії тим, що він може давати часткову, не всюди визначену функцію. Якщо при побудові функції дозволене використання

всіх трьох операторів, але функцію одержали всюду визначену, то цю функцію називають загальнорекурсивною. Якщо ж про область визначення нічого невідомо, або відомо, що функція не всюди визначена, то ця функція частково рекурсивна. Якщо при побудові функції використовувалося лише два оператори — суперпозиції і і примітивної рекурсії, то таку функцію називають примітивно рекурсивною. Примітивно рекурсивна функція завжди всюди визначена.

В мовній практиці тонкощами слововжитку часто нехтують і кажуть про рекурсивні функції — а про які саме, або видно із мовного оточення (контексту) або це не має значення.

Теза Чьорча: Всі обчислювані в інтуїтивному розумінні функції є частково рекурсивними.

Будемо користуватися тим, що всі функції на розширеному натуральному ряді, які можна задати арифметичними виразами в шкільному розумінні, є рекурсивними.

Після того, як ми вказали два підходи до формалізації інтуїтивної обчислюваності, і вказали дві тези — Тьюрінга і Чорча, можна говорити просто про обчислювані функції — а чи вони обчислювані на машині Тьюрінга, чи вони є частково рекурсивними функціями, чи для них є по іншому вказаний алгоритм обчислення, — все це не має принципового значення.

1.3 Перераховні множини

Область значень обчислюваної функції називається перераховною множиною. Якщо перераховна множина $M \subseteq \overline{\mathbb{N}}$ є множиною всіх значень обчислюваної функції f , то кажуть, що M перераховується функцією f .

Множина всіх натуральних чисел є перераховною множиною, тому що це область значень обчислюваної функції $s(x) = x + 1$. Весь розширений натуральний ряд є перераховною множиною, оскільки це область значень функції $f(x) = x$. В по-

передньому розділі ми розглянули частково рекурсивну функцію, у якої область визначення складається лише із 5, а область значень, лише із 0. Отже множина, що складається лише із 0, є перераховною.

Наявність загальнорекурсивних функцій f_{ev}, f_{odd} доводить, що множина парних і множина непарних чисел є перераховними.

Доведемо, що перетин і об'єднання двох перераховних множин є перераховними множинами. Спочатку доведемо першею

Терема. Перетин двох перераховних множин є перераховною множиною.

Доведення. Нехай множини $A, B \subseteq \bar{\mathbb{N}}$ є перераховними і перераховуються вони функціями $u(x), v(x)$ відповідно, тобто A є множиною значень функції $u(x)$ а B є множиною значень функції $v(x)$. Позначимо через $v^{-1}(x)$ функцію, яка одержується із функції $v(x)$ застосуванням оператора мінімізації. Таким чином $v(v^{-1}(x)) = x$, якщо x належить області значень функції $v(x)$. І функція $v^{-1}(x)$ не визначена в точці x , якщо x не належить області значень функції $v(x)$.

Розглянемо функцію

$$w(x) = v(v^{-1}(u(x))).$$

Вона є обчислюваною, оскільки є композицією обчислюваних функцій.

Якщо $y \in A \cap B$, то $v(v^{-1}(y)) = y$, (за визначенням функції $v^{-1}(x)$) і $y = u(x)$ для деякого x (за визначенням області значень функції u). І для так вибраного x буде

$$w(x) = v(v^{-1}(u(x))) = v(v^{-1}(y)) = y.$$

Ми довели, що коли $y \in A \cap B$, то y належить області значень функції.

Якщо число $w(x)$ визначене, то

$$v(v^{-1}(u(x))) = u(x). \quad (9)$$

Ліва частина рівності (9) показує, що область значень функції w міститься в B . А права частина рівності (9) показує, що область значень функції w міститься в A . Отже область значень функції w міститься в перетині $A \cap B$.

Доведення закінчене.

Доведемо, що об'єднання перераховних множин є перераховною множиною. З огляду на важливість твердження сформулюємо його у вигляді теореми.

Теорема. Об'єднання двох перераховних множин є перераховною множиною.

Доведення. Нехай множини $A, B \subseteq \bar{\mathbb{N}}$ є перераховними і перераховуються вони функціями $u(x), v(x)$ відповідно, тобто A є множиною значень функції $u(x)$ а B є множиною значень функції $v(x)$.

Користуємося тим, що функції

$$f_{\text{odd}}, \quad f_{\text{ev}}, \quad \frac{x}{2}, \quad x+1, \quad x-1, \quad x \dot{-} 1$$

є рекурсивними — це доведене вище. Для більшої наочності будемо використовувати табличне задання функції — двома рядками: у верхньому рядку пишемо значення аргумента, а в нижньому пишемо значення функції.

Застосовуємо оператор примітивної рекурсії до функцій $\phi(x_1) = 0 = 0(x_1)$, $\psi(x_1, x_2, x_3) = u(x_3)$. Одержану функцію позначимо через u_1 . Таким чином

$$u_1(n) = \begin{cases} 0, & \text{якщо } n = 0, \\ u(n-1), & \text{якщо } n \neq 0 \end{cases} \quad u_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & u(0) & u(1) & u(2) & u(3) & u(4) \end{pmatrix}$$

Суперпозицій функцій $\frac{x}{2}, f_{\text{ev}}(x)$ буде функція

$$\frac{f_{\text{ev}}(x)}{2} = \begin{cases} x, & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ 0, & \text{якщо } x = 2k+1 \text{ для деякого } k = 0, 1, \dots \end{cases} \quad \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & \dots \\ 0 & 0 & 1 & 0 & 2 & 0 & \dots \end{pmatrix}$$

Створюємо суперпозицію

$$u_1\left(\frac{f_{ev}(x)}{2}\right) = \begin{cases} u_1(x), & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ u_1(0), & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases}$$

В табличному записі маємо

$$u_1\left(\frac{f_{ev}}{2}\right) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & \dots \\ u_1(0) & u_1(0) & u_1(1) & u_1(0) & u_1(2) & u_1(0) & \dots \end{pmatrix} = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & u(0) & 0 \end{pmatrix}$$

Суперпозицією одержаної функції і функції $x+2$ буде функція

$$u_2 = u_1\left(\frac{f_{ev}(x+2)}{2}\right) = \begin{cases} u(k), & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ 0, & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases}$$

Подібним чином будуємо функцію

$$v_1(n) = \begin{cases} 0, & \text{якщо } n = 0, \\ v(n-1), & \text{якщо } n \neq 0 \end{cases}$$

і функцію

$$v_2 = v_1\left(\frac{f_{ev}(x+1)}{2}\right) = \begin{cases} 0, & \text{якщо } x = 2k, \text{ для деякого } k = 0, 1, \dots \\ v(k), & \text{якщо } x = 2k + 1 \text{ для деякого } k = 0, 1, \dots \end{cases}$$

Суперпозицією функцій $x_1 + x_2$ і u_2, v_2 буде функція

$$u_2(n) + v_2(n) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & \dots \\ u(0) & v(0) & u(1) & v(1) & u(2) & v(2) & \dots \end{pmatrix}$$

областю значень якої є $A \cup B$.

Теоретико-множинна різниця перераховних множин не обов'язково перераховна множина.

Доведення закінчене.

1.4 Обчислювані предикати.

Предикат $P(x_1, x_2, \dots, x_n)$ з вільними змінними $x_1, x_2, \dots, x_n$ називають обчислюваним, коли обчислюваною є функція — характеристична функція області істиності цього предиката

$$f(x_1, x_2, \dots, x_n) = \begin{cases} 1, & \text{якщо } P(x_1, x_2, \dots, x_n) = 1, \\ 0, & \text{якщо } P(x_1, x_2, \dots, x_n) = 0. \end{cases}$$

Кажуть, що предикат $P(x_1, x_2, \dots, x_n)$ обчислюється функцією f

Наведемо приклади. Одномісний предикат $z = 0$ обчислюваний, він обчислюється функцією

$$f(z) = \begin{cases} 1, & \text{якщо } z = 0, \\ 0, & \text{якщо } z \neq 0. \end{cases}$$

Двомісний предикат $x = y$ обчислюваний, він обчислюється функцією

$$g(x, y) = f((x \dot{-} y) + (y \dot{-} x)),$$

де функцію f взяли із попереднього прикладу.

Одномісний предикат $\exists x(2x = y)$ обчислюваний, він обчислюється функцією $f_3(y)$ (див. формулу (5))

Будемо вважати очевидним, що коли предикат P обчислюється функцією f , предикат Q обчислюється функцією g , то предикат $P \wedge Q$ обчислюється функцією $f \cdot g$, а предикат $P \vee Q$ обчислюється функцією $f + g - fg$.

Множина M називається обчислюваною, коли обчислюваним є предикат $x \in M$.

1.5 Конструктивний напрям в математиці

Вище відмічено, що існують необчислювані функції, предикати, множини. так ми уявляємо, що в кожній непорожній мно-

жині можна вибрати один елемент, але алгоритма, як вибрати цей елемент, немає. Виникає питання, а які твердження в математиці можна обґрунтувати, використовуючи лише ті засоби, які забезпечуються відповідними алгоритмами. Напрямок математики, в якому всі теореми проходять тестування цим питанням, називається конструктивним. Для математики, який сповідує конструктивний напрям, функція

$$f(z) = \begin{cases} 1, & \text{якщо } z = 0, \\ 0, & \text{якщо } z \neq 0. \end{cases}$$

на множині дійсних чисел не існує, тому що немає алгоритма, який би дозволив перевірити — задане дійсне число є нулем чи ні.

Число $\sqrt{2}$ існує, тому що можна вказати алгоритм послідовної побудови цифр в його десятковому записі. Взагалі, дійсні числа з конструктивної точки зору це алгоритми, що дозволяють знайти будь-яку (скінченну!) множину цифр в десятковому записі.

Увесь конструктивний напрям не однорідний — він розділяється на дві течії. Одна вимагає точного формулювання поняття алгоритм. Це так звана конструктивна математика. Основоположниками напрямку можна вважати творців частково рекурсивних функцій.

Друга течія користується інтуїтивним розумінням поняття алгоритм. Це так званий інтуїціонізм⁵. Відлік часу життя інтуїтивізму починається з 1907 року, з першої роботи Брауера (Brouwer)

⁵Не плутати з філософським напрямком — інтуїтивізмом

Показчик

- алгоритм, 1
 - інтуїтивний, 2
 - неформальний, 2
- функція
 - характеристична, 1
- функція
 - проектування, 4
- функція, 1
 - частково рекурсивна, 3
 - найпростіша, 4
 - обчислювана, 1
 - примітивно рекурсивна, 3
 - селективна, 4
 - тотожний нуль, 4
 - вибору, 4
 - загально рекурсивна, 3
 - значення, 1
 - знаходження наступного числа, 4
- машина
 - Тьюрінга, 2
- множини
 - перераховні, 11
- моделювання, 2
- оператор, 4
 - мінімізації, 4
 - примітивно рекурсії, 4
 - примітивно рекурсії, 5
 - суперпозиції, 4
- предикат
 - обчислюваний, 14
- ряд
 - розширений натуральний, 1
- схема
 - примітивно рекурсії, 5
- теза
 - Чорча, 10
 - Тьюрінга, 2
- змінна
 - фіктивна, 6