РЕКУРСИВНІ ФУНКЦІЇ.

Будь-який алгоритм однозначно ставить у відповідність до вхідних даних (у випадку, якщо він визначений на них) результат. Тому, як ми вже зазначали, з кожним алгоритмом однозначно зв'язана функція, яку він обчислює. Однак, обернене твердження не завжди вірне, тобто не для будь-якої функції існує алгоритм, який її обчислює. Виникає питання: для яких функцій алгоритми існують? Як описати такі ефективно обчислювані функції?

Дослідження цих питань призвело до створення в 30-х роках XX століття алгоритмічної системи, заснованої на використанні конструктивно визначаємих арифметичних (цілочислових) функцій, які одержали спеціальну назву *рекурсивні функції*. Зазначена алгоритмічна система історично була першою і перед тим, як ми розглянемо основні її твердження, наведемо кілька означень і тезисів, які лежать в основі даної алгоритмічної системи.

Рекурсією називається спосіб завдання функції, при якому значення обчислюваної функції для довільних значень аргументів виражається відомим чином через значення цієї функції для менших значень аргументів. Як видно, для рекурсивної функції існує певна ефективна процедура її обчислення.

Обчислюваними функціями називаються чисельні функції, значення яких також можна встановити за допомогою деякої ефективної процедури. Обчислювана функція задає відповідність між елементами множини X (область визначення функції) і елементами множини Y (сукупність значень функції). Якщо область визначення функції із X в Y збігається з множиною X, то функція називається *всюди визначеною*. Якщо тільки деяким елементам множини X поставлені у відповідність однозначно визначені елементи множини Y, то говорять, що задано *часткову функцію* з X в Y.

Поняття ефективної процедури в цих означеннях є інтуїтивним. Кажуть, що існує ефективна процедура для виконання певних обчислень, якщо ці обчислення виконуються за відповідними правилами, тобто згідно з деяким алгоритмом. Таким чином, ми звели поняття алгоритму до поняття обчислюваної функції. Але, незважаючи на те, що поняття алгоритму і поняття обчислюваної функції тут береться в інтуїтивному (не математичному) змісті, при переході від алгоритмів до обчислюваних функцій виникає одна дуже істотна обставина. Сукупність процесів, які підпадають під інтуїтивне поняття алгоритму, досить обширна і не піддається підрахунку, а сукупність обчислюваних функцій для реалізації алгоритмів (у найбільш загальному уявленні про них) завжди виявляється однією і тією ж та легко описується в рамках математичних теорій звичними для нас термінами. Однією з таких теорій є математична теорія рекурсивних функцій.

Теза про те, що поняття обчислюваної функції дійсно зводиться до строгого поняття рекурсивної функції, була висунута у працях таких відомих математиків, як Чёрч, Гёдель і Кліні. Гёдель вперше описав клас всіх рекурсивних функцій як клас всіх чисельних функцій, визначених у деякій формальній системі. Виходячи із зовсім інших передумов, Чёрч у 1936 році вивів той же клас чисельних функцій, що і Гёдель. Чёрчем була висунута гіпотеза про те, що *клас рекурсивних функцій тотожний класу всюди обчислюваних функцій*. Ця гіпотеза стала відома під ім'ям *тези Чёрча*. Кліні ввів поняття *частково рекурсивної функції* і висловив гіпотезу про те, що всі часткові обчислювані функції (тобто часткові функції, які обчислюються за допомогою алгоритму), є *частково рекурсивними*. Надалі під тезою Чёрча стали розуміти гіпотезу Чёрча в тому розширеному вигляді, яку придав їй Кліні: *сукупність числових функцій, яка збігається із сукупністю всіх (всюди визначених чи часткових) обчислюваних функцій при найбільш широкому дотепер відомому розумінні алгоритму, має назву сукупності рекурсивних функцій.*

Ми вже відзначали, що поняття обчислюваної функції точно (математично) не визначається, тому тезу Чёрча довести не можна. Однак математичні дослідження підтвердили доцільність вважати поняття частково рекурсивної функції науковим еквівалентом поняття обчислюваної часткової функції.

Теза Чёрча виявилась достатньою для того, щоб надати необхідну точність формулюванню алгоритмічних проблем і в ряді випадків зробити можливим доведення їхньої нерозв'язаності. Поняття алгоритму зводиться до поняття обчислюваної функції, а в силу тези Чёрча питання про обчислюваність функції рівносильне питанню про її рекурсивність. Оскільки поняття рекурсивної функції є строгим, то за допомогою звичайних математичних перетворень вдається іноді безпосередньо довести, що функція, яка розв’язує задачу, не може бути рекурсивною, тобто задача алгоритмічно нерозв’язна.

В теорії алгоритмів застосування рекурсивних функцій основане на ідеї нумерації слів у довільному алфавіті послідовними натуральними числами. Найпростіше таку нумерацію можна здійснити, розташовуючи слова в порядку зростання їхніх довжин, а слова, які мають однакову довжину, можна розташовувати у довільному (наприклад, лексикографічному) порядку. Після нумерації вхідних і вихідних слів у довільному алфавітному операторі цей оператор перетворюється у функцію $y = f(x)$, аргумент якої x, а також і сама функція y приймає цілочислові додатні значення. Функція $f(x)$ може бути визначена не для всіх, а тільки для деяких значень x, які складають її область визначення. Подібні частково визначені цілочислові функції для стислості звичайно називають *арифметичними функціями*.

Ми вже відзначали, що в теорії алгоритмів застосовують конструктивний (фінітний) підхід. При застосуванні до теорії рекурсивних функцій він полягає в тому, що множина досліджуваних об'єктів (функцій) будується зі скінченного числа початкових або базисних функцій за допомогою простих операцій, які просто та ефективно виконуються. Зазначені базисні функції прийнято ще називати *елементарними арифметичними функціями*.

1. Функція, яка тотожно дорівнює нулю:

$$\mathbf{O}^n(x_1, x_2, x_3, \ldots, x_n) = 0. \tag{1}$$

Функція (1) визначена для всіх цілих додатних значень аргументів.

2. Тотожна функція, яка повторює значення своїх аргументів:

$$\mathbf{I}_m^n(x_1, x_2, x_3, \ldots, x_n) = x_m. \tag{2}$$

Аналогічно функції (1), верхній індекс n тотожної функції (2) визначає число цілих додатних аргументів (слів) на який вона визначена, а нижній індекс m – номер аргументу, який тотожно повертається функцією. Таким чином, $1 \le m \le n$, а в частковому випадку $\mathbf{I}_1^1(x) = x$.

3. Функція безпосереднього слідування:

$$\mathbf{S}^1(x) = x + 1. \tag{3}$$

Функція безпосереднього слідування (3) також визначена для цілого додатного значення свого аргументу.

В теорії рекурсивних функцій особливе місце займають дві операції, які дозволяють будувати зі скінченного числа базисних функцій усе більш і більш складні нові арифметичні функції.

Достатньо потужним засобом, який дозволяє діставати вказані нові функції, є *операція суперпозиції функцій,* яка полягає в підстановці одних функцій замість аргументів інших арифметичних функцій.

Нехай задані n функцій від m змінних кожна: $f_1(x_1, x_2, x_3, \ldots, x_m)$,

$f_2(x_1, x_2, x_3, \dots, x_m)$, $f_3(x_1, x_2, x_3, \dots, x_m)$, …, $f_n(x_1, x_2, x_3, \dots, x_m)$. Крім того, задана функція від n змінних $f(x_1, x_2, x_3, \dots, x_n)$. Тоді суперпозиція функцій f_i, $i = 1, 2, 3, \dots, n$ з функцією f дасть нам нову, більш складну функцію від m змінних:

$$g(x_1, x_2, x_3, \dots, x_m) = f\begin{pmatrix} f_1(x_1, x_2, x_3, \dots, x_m), f_2(x_1, x_2, x_3, \dots, x_m), \\ f_3(x_1, x_2, x_3, \dots, x_m), \dots, f_n(x_1, x_2, x_3, \dots, x_m) \end{pmatrix}. \qquad (4)$$

Формула (4) визначає операцію суперпозиції в загальному вигляді. В частковому випадку, здійснимо, наприклад, суперпозицію функції, яка тотожно дорівнює нулю і функції безпосереднього слідування: $g(x) = \mathbf{S}^1(\mathbf{O}^1(x)) = 0 + 1 = 1$. При суперпозиції функції $\mathbf{S}^1(x)$ із самою собою одержимо функцію $g(x) = \mathbf{S}^1(\mathbf{S}^1(x)) = (x+1) + 1 = x + 2$, і т.п.

Операція суперпозиції звичайно позначається оператором $\mathbf{P}^{n+1}$, де верхній індекс необхідно розуміти, як кількість функції, що беруть участь у суперпозиції.

Другою операцією, яка дозволяє конструктивно будувати нові функції, є операція *примітивної рекурсії*. Вона дозволяє побудувати $(n+1)$-місну арифметичну функцію (функцію від $n+1$ аргументу) за двома заданими функціями: n-місної та $(n+2)$-місної.

Нехай задані які-небудь числові часткові функції: n-місна g та $(n+2)$-місна h. Кажуть, що $(n+1)$-місна часткова функція f *виникає примітивною рекурсією* з функцій g і h, якщо для всіх натуральних значень $x_1, x_2, x_3, \dots, x_n, y$ маємо:

$$\left.\begin{aligned} f(x_1, x_2, x_3, \dots, x_n, 0) &= g(x_1, x_2, x_3, \dots, x_n); \\ f(x_1, x_2, x_3, \dots, x_n, y+1) &= h(x_1, x_2, x_3, \dots, x_n, y, f(x_1, x_2, x_3, \dots, x_n, y)). \end{aligned}\right\}. \qquad (5)$$

Система рівнянь (5) називається *схемою примітивної рекурсії*. Операцію примітивної рекурсії скорочено позначають оператором $\mathbf{R}$ і символічно пишуть $f = \mathbf{R}(g, h)$.

У випадку, коли $n = 0$, тобто обчислювана функція f є одномісною, схема (5) приймає більш простий вигляд:

$$\left.\begin{aligned} f(0) &= const; \\ f(y+1) &= h(y, f(y)). \end{aligned}\right\} \qquad (6)$$

Рекурсивні схеми (5) і (6) визначають f не тільки через інші функції g і h, але і через значення f в попередніх точках: значення f в точці $y+1$ залежить від значення f в точці y.

Істотним в операторі примітивної рекурсії є те, що незалежно від числа змінних в f, рекурсія ведеться тільки за однією змінною y, а інші n змінних $x_1, x_2, x_3, \dots, x_n$ фіксуються, тобто відіграють роль параметрів. Наприклад, для обчислення $f(x_1, x_2, x_3, \dots, x_n, y = k)$ за схемою (5) буде необхідно виконати $k+1$ обчислення для відповідних значень $y = 0, 1, 2, \dots, k$:

$$\begin{aligned} &f(x_1, x_2, x_3, \dots, x_n, 0) = g(x_1, x_2, x_3, \dots, x_n); \\ &f(x_1, x_2, x_3, \dots, x_n, 1) = h(x_1, x_2, x_3, \dots, x_n, 0, g(x_1, x_2, x_3, \dots, x_n)); \\ &f(x_1, x_2, x_3, \dots, x_n, 2) = h(x_1, x_2, x_3, \dots, x_n, 1, f(x_1, x_2, x_3, \dots, x_n, 1)); \\ &\dots \\ &f(x_1, x_2, x_3, \dots, x_n, k) = h(x_1, x_2, x_3, \dots, x_n, k-1, f(x_1, x_2, x_3, \dots, x_n, k-1)). \end{aligned} \qquad (7)$$

Видно, що схема (7) дозволяє визначити функцію f однозначно, і якщо ми якимось чином вміємо визначати значення функцій g і h, то значення f можна обчислити за допомогою процедури цілком механічного характеру. Для визначення значення $f(a_1, a_2, a_3, \ldots, a_n, k)$ достатньо послідовно знайти числа:

$$z_0 = g(a_1, a_2, a_3, \ldots, a_n),\ z_1 = h(a_1, a_2, a_3, \ldots, a_n, 0, z_0),$$
$$z_2 = h(a_1, a_2, a_3, \ldots, a_n, 1, z_1), \ldots, z_k = h(a_1, a_2, a_3, \ldots, a_n, k-1, z_{k-1}).$$

Отримане на $(k+1)$-ому кроці число z_k і буде шуканим значенням функції f в точці $(a_1, a_2, a_3, \ldots, a_n, k)$.

Як приклад застосування оператора примітивної рекурсії покажемо, що за допомогою цієї операції із базисних функцій можна побудувати двомісну функцію підсумовування $f_+(x, y) = x + y$. Ця функція визначається за допомогою тотожної функції $g(x) = \mathbf{I}_1^1(x)$ і функції безпосереднього слідування $h(x, y, z) = \mathbf{S}^1(z)$. Справді, застосування операції $f_+ = \mathbf{R}(\mathbf{I}_1^1(x) = x, \mathbf{S}^1(z) = z+1)$ дає:

$$\begin{aligned}
&f(x, 0) = g(x) = x;\\
&f(x, 1) = h(x, 0, x) = x+1;\\
&f(x, 2) = h(x, 1, x+1) = (x+1)+1 = x+2;\\
&\ldots\\
&f(x, y-1) = h(x, y-2, x+(y-2)) = (x+(y-2))+1 = x+y-1;\\
&f(x, y) = h(x, y-1, x+(y-1)) = (x+(y-1))+1 = x+y.
\end{aligned}$$

Аналогічно можна побудувати й інші відомі арифметичні функції.

Функції, які можуть бути побудовані з елементарних арифметичних функцій за допомогою операцій суперпозиції і примітивної рекурсії, застосованих будь-яке скінченне число раз у довільній послідовності, називаються *примітивно рекурсивними функціями*. Операції суперпозиції і примітивної рекурсії, будучи застосовані до всюди визначених функцій, дають в результаті знову всюди визначені функції.

Означенню примітивної рекурсивності функцій можна надати більш формальний індуктивний вигляд.

1. Функції $\mathbf{O}^n(x_1, x_2, x_3, \ldots, x_n) = 0$, $\mathbf{I}_m^n(x_1, x_2, x_3, \ldots, x_n) = x_m$ і $\mathbf{S}^1(x) = x+1$ для всіх натуральних n і m, де $1 \le m \le n$, є примітивно рекурсивними.

2. Якщо $f_1(x_1, x_2, x_3, \ldots, x_m)$, $f_2(x_1, x_2, x_3, \ldots, x_m)$, $f_3(x_1, x_2, x_3, \ldots, x_m)$, …, $f_n(x_1, x_2, x_3, \ldots, x_m)$ і $f(x_1, x_2, x_3, \ldots, x_n)$ – примітивно рекурсивні функції, то $\mathbf{P}^{n+1}$ – примітивно рекурсивна функція для всіх натуральних n і m.

3. Якщо $g(x_1, x_2, x_3, \ldots, x_n)$ і $h(x_1, x_2, x_3, \ldots, x_n, y, z)$ примітивно рекурсивні функції, то $\mathbf{R}(g, h)$ примітивно рекурсивна функція.

4. Інших примітивно рекурсивних функцій немає.

З такого індуктивного опису можна витягти процедуру доведення примітивної рекурсивності (обчислюваності) функцій.

<u>Двомісна функція додавання</u>: $f_+(x, y) = x + y$.

Ця функція задовольняє співвідношенням:

$f_+(x, 0) = x + 0 = x$ – тотожна функція $\mathbf{I}_1^1(x)$;
$f_+(x, y+1) = x + (y+1) = (x+y)+1 = f_+(x, y)+1$ – функція безпосереднього слідування $\mathbf{S}^1(f_+(x, y))$.

Таким чином, функція $f_+(x, y)$ примітивно рекурсивна, оскільки виникає з базисних функцій $g(x)=x$ і $h(x, y, z)=z+1$ операцією примітивної рекурсії.

Порушимо питання: чи усі функції є примітивно рекурсивними? Виявляється, що ні. Так, наприклад, у теорії множин доведено, що множина усіх функцій $N \to N$ (одномісні цілочислові функції) є незчисленною. Тим більше незчисленною буде множина функцій типу $N^n \to N$. Кожна примітивно рекурсивна функція має скінченний опис, тому що вона задається скінченним словом у деякому фіксованому для усіх функцій алфавіті. Оскільки множина усіх скінченних слів зчисленна, то примітивно рекурсивні функції утворюють не більш ніж зчислену підмножину незчисленної множини функцій типу $N^n \to N$. Однак, строго кажучи , ми маємо справу не з самими функціями, з їхнім примітивно рекурсивним описом, отже, необхідно ставити більш вузьке питання: чи всі обчислювані функції можна описати як примітивно рекурсивні? У теорії алгоритмів на прикладі функції Аккермана доведено, що це не так. Ми ж без доведення відзначимо, що функція Аккермана $f(k)=a^{a^{\cdot^{\cdot^{a}}}\}k \text{ раз}}$, росте швидше, ніж будь-яка примітивно рекурсивна функція, і, отже, вона не є примітивно рекурсивною.

Можна зробити висновок про те, що більшість арифметичних функцій відносяться до примітивно рекурсивних, однак множина примітивно рекурсивних функцій не охоплює всіх арифметичних функцій, які можуть бути визначені конструктивно. Для розширення засобів побудови обчислюваних функцій найбільш придатною є *операція найменшого кореня (операція мінімізації).*

Операція найменшого кореня дозволяє визначити нову арифметичну функцію $f(x_1, x_2, x_3, \ldots, x_n)$ від n змінних за допомогою раніше побудованої функції $g(x_1, x_2, x_3, \ldots, x_n, y)$ від $(n+1)$-ої змінної. Для будь-якого заданого набору значень змінних $x_1=a_1$, $x_2=a_2$, $x_3=a_3$, …, $x_n=a_n$ як відповідне значення $f(a_1, a_2, a_3, \ldots, a_n)$ приймається найменший цілий невід'ємний корінь $y=b$ рівняння $g(a_1, a_2, a_3, \ldots, a_n, y)=0$. У випадку неіснування цілих невід'ємних коренів у цього рівняння, функція $f(x_1, x_2, x_3, \ldots, x_n)$ вважається невизначеною при відповідному наборі значень змінних. Звичайно приймається також, що функція $f(x_1, x_2, x_3, \ldots, x_n)$ не визначена на наборі $x_1=a_1$, $x_2=a_2$, $x_3=a_3$, …, $x_n=a_n$, якщо при існуванні найменшого кореня рівняння хоча б при одному цілому невід'ємному значенні $y=c$, яке задовольняє співвідношенню $0 \le c \le b-1$, функція виявляється невизначеною.

Арифметичні функції, які можуть бути побудовані з елементарних арифметичних функцій за допомогою операцій суперпозиції, примітивної рекурсії і найменшого кореня, називаються *частково рекурсивними функціями*. Якщо такі функції виявляються до того ж всюди визначеними, то вони називаються *загальнорекурсивними*.

У цьому означенні, як і у означенні примітивно рекурсивних функцій, передбачається можливість виконання всіх припустимих операцій у будь-якій послідовності та будь-яке скінченне число раз.

Частково рекурсивні функції зображують найбільш загальний клас конструктивно визначаємих арифметичних функцій.

Поняття частково рекурсивної функції є одним з основних понять теорії алгоритмів. Значення його полягає в наступному:

1. Кожна стандартно задана частково рекурсивна функція обчислюється шляхом відповідної процедури механічного характеру.

2. Які б класи точно описаних алгоритмів дотепер фактично не будувалися, у всіх випадках незмінно виявлялося, що числові функції, які обчислюються за допомогою алгоритмів цих класів, були частково рекурсивними. Тому загальноприйнятою є природничо-наукова гіпотеза Чёрча, про яку ми говорили раніше. Гіпотеза (теза) Чёрча

дає алгоритмічне тлумачення поняттю частково рекурсивних функцій.

Практично поняттям частково рекурсивних функцій користуються для доведення алгоритмічної розв'язаності чи нерозв'язаності проблем. Використання частково рекурсивних функцій для зображення того чи іншого конкретного алгоритму практично недоцільне через складність такого процесу алгоритмізації.