Модуль 7.2. Частично рекурсивные функции

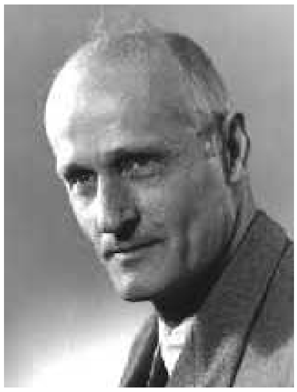


Рис. 1 – Стивен Клини

Мы познакомились с неформальными определениями алгоритма и вычислимой функции. Но для математического изучения этих понятий вычислимость следует формализовать. В 30-е годы XX века и позже было предложено несколько точных определений понятия алгоритма и вычислимой функции. Опишем подход Гёделя и Клини[1] (рис. 1), предложенный ими в 1936 г.

Основная идея состояла в том, чтобы получить все вычислимые функции из существенно ограниченного множества базисных функций с помощью простейших алгоритмических средств.

Множество *исходных* функций таково:

- постоянная функция $0(\boldsymbol{x}) = 0$;
- одноместная функция следования $s(x) = x + 1$;
- функция проекции pr_i, $1 \leqslant i \leqslant k$, $pr_i(\boldsymbol{x}) = x_i$.

Нетривиальные вычислительные функции можно получать с помощью композиции (суперпозиции) уже имеющихся функций. Этот способ явно алгоритмический.

1. **Оператор суперпозиции**. Говорят, что k-местная функция $f(\boldsymbol{x})$ получена с помощью суперпозиции из m-местной функции $\varphi(y_1, y_2, \ldots, y_m)$ и k-местных функций $g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})$, если $f(\boldsymbol{x}) = \varphi\big(g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\big)$.

   Второй (несколько более сложный) способ действует так.

2. **Примитивная рекурсия**. При $n \geqslant 0$ из n-местной функции f и $(n+2)$-местной функции g строится $(n+1)$-местная функция h по следующей схеме:

   - $h(\boldsymbol{x}, 0) = f(\boldsymbol{x})$,
   - $h(\boldsymbol{x}, y+1) = g\big(\boldsymbol{x}, y, h(\boldsymbol{x}, y)\big)$.

   При $n = 0$ получаем (a — константа):

   - $h(0) = a$;
   - $h(y+1) = g\big(y, h(y)\big)$.

   Два упомянутых способа позволяют задать только всюду определенные функции. Частично-определенные функции порождаются с помощью третьего гёделева механизма.

3. **Оператор минимизации**. Эта операция ставит в соответствие частичной функции $f\colon \boldsymbol{N}^{k+1} \to \boldsymbol{N}$ частичную функцию $h\colon \boldsymbol{N}^k \to \boldsymbol{N}$, которая определяется так ($\boldsymbol{x} = <x_1, \ldots, x_k>$):

[1] Стивен Коул Клини (1909–1994 гг.) — американский математик, логик.

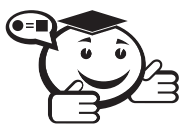
- область определения $D_h = \{\boldsymbol{x} \mid$ существует $x_{k+1} \geqslant 0,\ f(\boldsymbol{x}, x_{k+1}) = 0$ и $<\boldsymbol{x}, y> \in D_f$ для всех $y \leqslant x_{k+1}\}$;
- $h(\boldsymbol{x})$ = наименьшее значение y, при котором $f(\boldsymbol{x}, y) = 0$.

Оператор минимизации обозначается так: $h(\boldsymbol{x}) = \mu y[f(\boldsymbol{x}, y) = 0]$. Очевидно, что даже если f всюду определено, но нигде не обращается в 0, то $\mu y[f(\boldsymbol{x}, y) = 0]$ нигде не определено. Естественный путь вычисления $h(\boldsymbol{x})$ состоит в подсчете значения $f(\boldsymbol{x}, y)$ последовательно для $y = 0, 1, 2, \ldots$ до тех пор, пока не найдется y, обращающее $f(\boldsymbol{x}, y)$ в 0. Этот алгоритм не остановится, если $f(\boldsymbol{x}, y)$ нигде не обращается в 0.

В общих чертах роль оператора минимизации состоит во введении функций, заданных «неявно». Кроме того, минимизация позволяет вводить в вычисление перебор объектов для отыскания объекта в бесконечном семействе. Важно отметить две особенности оператора минимизации.

Выбор минимального числа y, для которого $f(\boldsymbol{x}, y) = 0$ требуется для обеспечения однозначности функции h.

Область определения функции h: на первый взгляд представляется искусственно суженной: если, скажем, $f(\boldsymbol{x}, 1) = 0$, а $f(\boldsymbol{x}, 0)$ не определено, мы считаем функцию $h(\boldsymbol{x})$ неопределенной, а не равной 1. Причина этого состоит в желании сохранить интуитивную вычислимость функции h.



Все функции, которые можно получить из базисных функций за конечное число шагов только с помощью трех указанных механизмов, называются ***частично рекурсивными****. Если функция получается всюду определенная, то тогда она называется* ***общерекурсивной****. Если функция получена без механизма минимизации, то в этом случае она называется* ***примитивно рекурсивной****.*

Любую примитивно рекурсивную функцию можно вычислить с помощью цикла в форме *for*, так как верхнюю границу для числа повторений можно указать заранее. Оператор минимизации позволяет описать функции, которые нельзя вычислить за заранее ограниченное число итераций, для вычисления их значений требуется цикл в форме *while*.

Можно легко показать [1, с. 28], что введение фиктивных переменных, а также перестановка и отождествление переменных не выводят за пределы класса примитивно рекурсивных функций и класса частично рекурсивных функций. Это проще всего объяснить на примерах.

Введение фиктивных переменных. Если $g(x_1, x_3)$ — примитивно рекурсивная функция и $f(x_1, x_2, x_3) = g(x_1, x_3)$, то $f(x_1, x_2, x_3)$ — примитивно рекурсивная функция.

Перестановка переменных. Если $g(x_1, x_2)$ — примитивно рекурсивная функция и $f(x_2, x_1) = g(x_1, x_2)$, то f есть также примитивно рекурсивная функция.

Отождествление переменных. Если $g(x_1, x_2, x_3)$ — примитивно рекурсивная функция и $f(x_1, x_2) = g(x_1, x_2, x_1)$, то $f(x_1, x_2)$ есть также примитивно рекурсивная функция.

Рассмотрим примеры частично рекурсивных функций. Все эти примеры и много других можно найти в [1, 2].

Пример 1

Примеры рекурсивности

Сложение двух чисел

sum: $<x, y> \to x + y$.

Эта функция является общерекурсивной в силу примитивной рекурсии

$sum(x, 0) = pr_1(x) = x$,

$sum(x, y+1) = s\big(sum(x, y)\big) = sum(x, y) + 1$.

Считая известным частичную рекурсивность функции sum, легко убедиться с помощью примитивной рекурсии и композиции в частичной рекурсивности функции $(x_1, x_2, \ldots, x_n) \to x_1 + x_2 + \ldots + x_n$.

Умножение двух чисел

$prod$: $<x, y> \to xy$.

Используем примитивную рекурсию

$prod(x, 0) = 0(x) = 0$,

$prod(x, y+1) = sum(prod(x, y), x)$.

Считая известным частичную рекурсивность функции $prod$, легко убедиться с помощью примитивной рекурсии и композиции в частичной рекурсивности функции $(x_1, x_2, \ldots, x_n) \to x_1 \cdot x_2 \cdot \ \ldots \cdot x_n$.

Усеченное вычитание 1

$\delta(x) = x - 1$, если $x > 0$,

$\delta(0) = 0$.

Эта функция примитивно рекурсивна, действительно,

$\delta(0) = 0 = 0(x)$,

$\delta(y+1) = y = pr_2(<x, y>)$.

Усеченная разность

$x \div y = x - y$, если $x \geqslant y$,

$x \div y = 0$, если $x < y$.

Эта функция примитивно рекурсивна, действительно,

$x \div 0 = x$,

$x \div (y+1) = \delta(x \div y)$.

Модуль разности

$|x - y| = x - y$, если $x \geqslant y$,

$|x - y| = y - x$, если $x < y$.

Эта функция примитивно рекурсивна в силу суперпозиции

$|x - y| = (x \div y) + (y \div x)$.

Факториал

Действительно,

$0! = 1$,

$(y+1)! = prod(y!, y+1)$.

$\min(x, y)$ — *наименьшее из чисел x и y.*

В силу суперпозиции: $\min(x, y) = x \div (x \div y)$.

Знак числа
$sg(x) = 0$, если $x = 0$,
$sg(x) = 1$, если $x > 1$.
В силу рекурсии
$sg(0) = 0$,
$sg(y + 1) = 1$.
$rm(x, y)$ — *остаток от деления* y *на* x, *если* $x \neq 0$, *и* y, *если* $x = 0$.
В силу рекурсии и суперпозиции
$rm(x, 0) = 0$,
$rm(x, y + 1) = prod\Big(s\big(rm(x, y)\big),\ sg\big(\left|x - s\big(rm(x, y)\big)\right|\big)\Big)$.

. . . .

Используя функции, для которых уже установлено, что они являются частично рекурсивными, мы получаем все новые и новые частично рекурсивные функции. Существуют критерии, которые позволяют установить частичную рекурсивность сразу для обширных классов функций (см., например, [2, с. 135–150]).

Используя минимизацию (μ-оператор), можно получать частично определенные функции из всюду определенных функций.

Пример 2

Пусть $f(x, y)$ есть частично рекурсивная функция $|x - y^2|$, тогда $g(x) = \mu y\big[f(x, y) = 0\big]$ — не всюду определенная функция:

$$g(x) = \sqrt{x},$$

если x есть точный квадрат, и неопределенна в противном случае.

. . . .

Таким образом, тривиально используя μ-оператор вместе с суперпозицией и рекурсией, можно построить больше функций, исходя из основных, чем только с помощью суперпозиции и рекурсии (так как эти операции порождают из всюду определенных функций всюду определенные). Существуют, однако, и общерекурсивные (всюду определенные) функции, для построения которых нельзя обойтись без минимизации.

Приведем пример функции, не являющейся примитивно рекурсивной, хотя и вычислимой в интуитивном смысле.

Определим последовательность одноместных функций F_n: $\mathbf{N} \to \mathbf{N}$, $n \in \mathbf{N}$, следующим образом:
$F_0(x) = x + 1$,
$F_{n+1}(x) = F_n\Big(F_n\big(\ldots F_n(1)\ldots\big)\Big)$ (F_n повторяется $x + 1$ раз).
Поэтому $F_1(x) = x + 2$, $F_2(x) = 2x + 3$, $F_3(x) = 2^{x+3} - 3$,
$F_4(x) = 2^{2^{\cdot^{\cdot^{2}}}} - 3$ (башня из $x + 3$ двойки) и т. д.

Имеем следующие свойства:

1) для каждого n функция $x \to F_n(x)$ является примитивно рекурсивной;
2) $F_n(x) > 0$;
3) $F_n(x+1) > F_n(x)$;
4) $F_n(x) > x$;
5) $F_{n+1}(x) \geqslant F_n(x+1)$;
6) для каждой k-местной примитивно рекурсивной функции $f(x_1, x_2, \ldots, x_k)$ существует такое n, что F_n мажорирует f, т. е.

$$f(x_1, x_2, \ldots, x_k) \leqslant F_n\big(\max(x_1, x_2, \ldots, x_k)\big)$$

   для всех $x_1, x_2, \ldots, x_k$;
7) функция $A(n, x) = F_n(x)$ не является примитивно рекурсивной [3, с. 53] (эта функция известна как ***функция Аккермана***[1]).

Функцию Аккермана можно определить и в традиционной записи:

- $f(0, y) = y + 1$,
- $f(x+1, 0) = f(x, 1)$,
- $f(x+1, y+1) = f\big(x, f(x+1, y)\big)$.

Позднее мы приведем доводы в пользу правдоподобности того, что понятие частично рекурсивной функции есть точный математический эквивалент интуитивной идеи (эффективно) вычислимой функции.

Список литературы по модулю

[1] Манин Ю. И. Вычислимое и невычислимое / Ю. И. Манин. — М. : Советское радио, 1980. — 128 с.

[2] Мендельсон Э. Введение в математическую логику / Э. Мендельсон. — М. : Наука, 1976. — 320 с.

[3] Катленд Н. Вычислимость. Введение в теорию рекурсивных функций : пер. с англ. / Н. Катленд. — М. : Мир, 1983. — 256 с.

[1]Вильгельм Фридрих Аккерман (1896–1962 гг.) — немецкий математик и логик.