

# Рекурсивные функции

**Эта модель рассматривает алгоритм как способ формирования одних вычислимых функций из других, т.е. одни функции конструктивно определяются из других.**

### *Понятие простейших функций*

Числовые функции, значение которых можно установить посредством некоторого алгоритма, называются вычислимыми функциями.

Для того чтобы описать класс функций с помощью рекурсивных определений, рассмотрим набор простейших функций:

1)  $Z(x_1, x_2, \dots, x_n) = 0$  - нуль-функция, которая определена для всех неотрицательных значений аргумента;

2)  $s(x) = x+1$  - функция непосредственного следования, также определенная для всех целых неотрицательных значений своего аргумента;

3)  $I_m^n(x_1, x_2, \dots, x_m, \dots, x_n) = x_m$  - функция выбора (тождества), повторяющая значения своих аргументов.

Используя простейшие функции в качестве исходных функций, можно с помощью небольшого числа общих конструктивных приемов строить сложные арифметические функции. В теории рекурсивных функций особо важное значение имеют три операции: суперпозиции, примитивной рекурсии и минимизации.

***Эти операторы сохраняют вычислимость.***

Все элементарные функции - всюду определенные и алгоритмически вычислимые.

# Оператор суперпозиции

Оператором суперпозиции  $S$  называется подстановка в функцию от  $m$  переменных  $m$  функций от  $n$  одних и тех же переменных. Она дает новую функцию от  $n$  переменных. Например, из функций  $f(x_1, x_2, \dots, x_m), f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n)$  можно получить новую функцию:

$$S^{m+1}(f, f_1, f_2, \dots, f_m) = g(x_1, x_2, \dots, x_n) = f(f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_m(x_1, \dots, x_n)).$$

(1)

В операции суперпозиции  $S^{m+1}$  индекс сверху указывает на число функций.

Таким образом, при помощи оператора суперпозиции и функции выбора можно выразить любую подстановку функции в функцию.

## Свойства операции суперпозиции.

1. Операция суперпозиции сохраняет свойство всюду определенности функций, т.е. если  $f(x_1, x_2, \dots, x_m), f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n)$  всюду определены, то и  $g(x_1, x_2, \dots, x_n)$  всюду определена.

2. Операция суперпозиции сохраняет свойство алгоритмической вычислимости функций.

# Примеры

Суперпозиция функций:

$f(x) = 0$  и  $g(x) = x+1$  получим функцию:  $h(x) = g(f(x)) = 0 + 1 = 1$

При суперпозиции функции  $g(x)$  с этой же функцией получим функцию  $h(x) = g(g(x)) = x + 2$ .

Используя подстановку и функции тождества, можно переставлять и отождествлять аргументы в функции:

$$f(x_2, x_1, x_3, \dots, x_n) = f(I_2^2(x_1, x_2), I_1^2(x_1, x_2), x_3, \dots, x_n);$$

$$f(x_1, x_1, x_3, \dots, x_n) = f(I_1^2(x_1, x_2), I_1^2(x_1, x_2), x_3, \dots, x_n).$$

Таким образом, если заданы функции тождества и операторы суперпозиции, то можно считать заданными всевозможные операторы подстановки функций в функции, а также переименования, перестановки и отождествления переменных.



# Оператор примитивной рекурсии

Оператор примитивной рекурсии задается следующим образом:

Рекурсия ведется по одному аргументу, все остальные считаются параметрами. Известны две функции:  $g(x_1, x_2, \dots, x_n)$  и  $h(x_1, x_2, \dots, x_n, y, z)$ .

**Определение.** Говорят, что функция  $f(x_1, \dots, x_n, y)$  получена из функцией  $g(x_1, \dots, x_n)$  и  $h(x_1, \dots, x_n, y, z)$  с помощью операции примитивной рекурсии, если выполняются следующие равенства:

$$\begin{aligned} f(x_1, \dots, x_n, 0) &= g(x_1, \dots, x_n) \\ f(x_1, \dots, x_n, y + 1) &= h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)). \end{aligned}$$

Это определение имеет смысл, когда  $n \neq 0$ , при этом записывается

$$f(x_1, \dots, x_n, y) = R(g(x_1, \dots, x_n), h(x_1, \dots, x_n, y, z))$$

или сокращенно

$$f = R(g, h),$$

где  $R$ —означает операции примитивной рекурсии.

В нуле функция  $f$  равна функции  $g$ , а в некоторой точке  $y+1$  она определяется через известную функцию  $h$  и значение этой же функции в предыдущей точке.

Оператор примитивной рекурсии  $R_n$  позволяет определить  $(n+1)$ - местную функцию  $f$  по двум заданным функциям, одна из которых является  $n$ - местной функцией  $g$ , а другая  $(n+2)$ - местной функцией  $h$ .

Приведенная пара равенств называется **схемой примитивной рекурсии**.

# Оператор примитивной рекурсии

## Основные свойства операции примитивной рекурсии.

Операция примитивной рекурсии, так же как и операция суперпозиции, сохраняет свойство всюду определенности и алгоритмической вычислимости. т.е. если  $g(x_1, \dots, x_n)$  и  $h(x_1, \dots, x_n, y, z)$  всюду определенные и вычислимые функции, то  $f(x_1, \dots, x_n, y)$  тоже будет всюду определенной функцией, где  $f = R(g, h)$ .

Всякая ПРФ является всюду определенной функцией.  
Всякая ПРФ является алгоритмически вычислимой.

# Оператор примитивной рекурсии

- Функция называется **примитивно – рекурсивной**, если она является элементарной или может быть получена из элементарных функций с помощью конечного числа применений операторов тождества, суперпозиции и примитивной рекурсии.
- Если некоторые функции являются примитивно-рекурсивными, то в результате применения к ним операторов суперпозиции или примитивной рекурсии можно получить новые примитивно-рекурсивные функции.
- Существует три возможности доказательства того, что функция является примитивно-рекурсивной:
  - а) показать, что заданная функция является простейшей;
  - б) показать, что заданная функция построена с помощью оператора суперпозиции;
  - в) показать, что заданная функция построена с помощью оператора примитивной рекурсии.

# Примеры доказательства вычислимости функций

## 1. Функция – константа

$$f(x) = m \quad s(s(s\dots s(Z(x))\dots)) \quad m\text{-раз}$$

## 2. Сложение

$$f(x,y)=x+y$$

$$f(x,0)=x;$$

$$f(x,y+1)=x+(y+1)=(x+y)+1=f(x,y)+1$$

**Доказательство:**

- $f(x,0)=g(x)=x=l(x);$
- $f(x,y+1) = h(x,y,z) = h(x,y,f(x,y)) = s(l_3^3(x,y,f(x,y)))$   
 $+^2 = R(l_1^1, [s^1; l_3^3]).$



### 3. Умножение

$$f(x,y)=x*y$$

$$f(x,0)=x*0=0;$$

$$f(x,y+1)=x*(y+1)=x*y+x=f(x,y)+x$$

**Доказательство:**

- $f(x,0)=g(x)=0=Z(x);$
- $f(x,y+1) = h(x,y,z) = h(x,y,f(x,y)) = x+z = I_3^1(x,y,f(x,y)) + I_3^3(x,y,f(x,y))$

$$x^2 = R(Z, [+; I_3^3, I_1^3])$$

### 4. Симметрическая разность (абсолютная величина разности)

$$x \dot{-} y = \begin{cases} x - y, & \text{если } x \geq y \\ 0, & \text{если } x < y \end{cases}$$

**Одноместная функция усеченного вычитания единицы**

определяется рек  $f(0) = 0 \dot{-} 1 = 0$

$$f(x+1) = x+1 \dot{-} 1 = x$$

$$\dot{-} 1 = R(Z, I_1^2)$$

$$f(x,0) = x \dot{\div} 0 = x$$

$$f(x,y+1) = x \dot{\div} (y+1) = (x \dot{\div} y) \dot{\div} 1 = f(x,y) \dot{\div} 1$$

**Доказательство:**

$$f(x,0) = g(x) = x = I(x);$$

$$f(x,y+1) = I_3^3(x,y,f(x,y)) \dot{\div} 1$$

$$\dot{\div}^2 = R(I_1^1, [\dot{\div} 1^1 I_3^3])$$

# Операции конечного суммирования и конечного произведения

**Определение.** Говорят, что функция  $\sigma(x_1, \dots, x_n, z)$  получена из функции  $g(x_1, \dots, x_n, y)$  с применением операции конечного суммирования, если для любого набора переменных  $(x_1, \dots, x_n, z)$  выполняется следующее равенство:

$$\begin{aligned} \sigma(x_1, \dots, x_n, z) &= g(x_1, \dots, x_n, 0) + g(x_1, \dots, x_n, 1) + \dots + g(x_1, \dots, x_n, z) = \\ &= \sum_{t=0}^z g(x_1, \dots, x_n, t) \end{aligned} \quad (1)$$

**Определение.** Говорят, что функция  $\delta(x_1, \dots, x_n, z)$  получена из функции  $g(x_1, \dots, x_n, y)$  с применением операции конечного произведения, если для любого набора переменных  $(x_1, \dots, x_n, z)$  выполняется следующее равенство:

$$\begin{aligned} \delta(x_1, \dots, x_n, z) &= g(x_1, \dots, x_n, 0) \cdot g(x_1, \dots, x_n, 1) \cdot \dots \cdot g(x_1, \dots, x_n, z) = \\ &= \prod_{t=0}^z g(x_1, \dots, x_n, t) \end{aligned} \quad (2)$$

**Теорема 1.** Операции конечного суммирования и конечного произведения сохраняют свойство примитивной рекурсивности функции.

# Оператор минимизации

Операция минимизации. Пусть задана  $n$ -местная функция  $g(x_1, \dots, x_{n-1}, x_n)$ . Зафиксируем набор  $(x_1, \dots, x_{n-1}, x_n)$  и рассмотрим уравнение относительно  $y$ :

$$g(x_1, \dots, x_{n-1}, y) = x_n \quad (*)$$

Будем решать данное уравнение, вычисляя последовательно

$g(x_1, \dots, x_{n-1}, 0), g(x_1, \dots, x_{n-1}, 1), g(x_1, \dots, x_{n-1}, 2)$  и сравнивая с  $x_n$ . Наименьшее  $y$ , для которого выполнено  $(*)$  обозначим

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

При этом считаем, что  $y$  определено, если  $g(x_1, \dots, x_{n-1}, z)$  определено при всех  $z \leq y$ . В противном случае считаем, что  $y$  неопределено.

Скажем, что функция  $f^n(x_1, x_2, \dots, x_n)$  получена с помощью оператора минимизации ( $\mu$ -оператора) из функции  $g^{n+1}(x_1, x_2, \dots, x_n, y)$ , если  $f^n(x_1, x_2, \dots, x_n)$  определена и равна  $y$  тогда и только тогда, когда все значения  $g^{n+1}(x_1, \dots, x_n, 0), \dots, g^{n+1}(x_1, \dots, x_n, y-1)$  определены и не равны 0, а  $g^{n+1}(x_1, \dots, x_n, y) = 0$ . В этом случае будем писать

$$f^n(x_1, \dots, x_n) = \mu y [g^{n+1}(x_1, \dots, x_n, y) = 0]$$



# Использование оператора минимизации

Используя минимизацию можно получать частично – определенные функции из всюду определенных

Пример 1.

$$g(x, y) = |x - 2y|$$

$$f(x) = \mu y [g(x, y) = 0] = \mu y [|x - 2y| = 0]$$

$$\text{Т.е. } f(x) = \mu y [|x - 2y| = 0] = x/2$$

Очевидно, что функция  $f$  определена только на числах вида  $2k$ ,  $k = 0, 1, 2, \dots$  ;  
и для каждого из них  $f(2k) = k$ .

**Пример 2.** Пусть  $g(x) = [x/2]$ . Найдем функцию  $f(x)$ , которая получается в результате применения оператора минимизации к этой всюду определенной функции. При каждом конкретном  $x$  значение  $f(x)$  равно минимальному корню уравнения  $[y/2] = x$ . Это уравнение имеет два корня:  $2x$  и  $2x+1$ . Поэтому  $f(x) = 2x$ .

**Пример 3.** Пусть требуется применить оператор минимизации по аргументу  $x_2$  к функции  $g(x_1, x_2) = x_1 \div x_2$ . Значения искомой функции  $f(x_1, x_2)$  будем получать отдельно для случаев  $x_2 = 0$  и  $x_2 > 0$ .

Пусть  $x_2 = 0$ , тогда уравнение (\*) будет иметь вид  $x_1 \div y = 0$ . Решая его относительно  $y$ , получаем бесконечное множество корней  $\{x_1, x_1 + 1, x_1 + 2, \dots\}$ . Поскольку минимальным корнем является  $x_1$ , то при всех  $x_1$  значение  $f(x_1, 0) = x_1$ .

Пусть теперь  $x_2 > 0$ . Тогда получаем уравнение  $x_1 \div y = x_2$ , которое при  $x_2 > 0$  равносильно уравнению  $x_1 - y = x_2$ . Последнее уравнение имеет единственное решение  $y = x_1 - x_2$ , если  $x_1 \geq x_2$ , либо вообще не имеет корней, если  $x_1 < x_2$ . В итоге получаем следующую функцию:

$$f(x_1, x_2) = \begin{cases} x_1 - x_2, & \text{если } x_1 \geq x_2, \\ \text{не определено иначе.} \end{cases}$$

**Пример 4.**

Пусть  $f(x, y) = |x - y^2|$ . Определим  $\varphi(x) = \mu y [|x - y^2| = 0]$ .

$\varphi(x) = \sqrt{x}$ , если  $x$  точный квадрат и неопределенна в противном случае.

# Тезис Черча

Функция называется **частично-рекурсивной (вычислимой по Черчу)**, если она может быть получена из простейших функций с помощью конечного числа операторов суперпозиции, примитивной рекурсии и минимизации.

Если такие функции оказываются всюду определенными, то они называются **общерекурсивными функциями**.

Указанные операции могут быть выполнены в любой последовательности и любое конечное число раз. Таким образом, мы не просто задаем функцию, но и точно знаем, как её вычислять.

Очевидно, каждая примитивно рекурсивная функция является частично рекурсивной, но обратное неверно.

Введем обозначения:

$K_{\text{ПРФ}}$  – класс примитивно рекурсивных функций;

$K_{\text{ОРФ}}$  – класс общерекурсивных функций;

$K_{\text{ЧРФ}}$  – класс частично рекурсивных функций.

Тогда между этими классами имеется соотношения:

$$K_{\text{ПРФ}} \subseteq K_{\text{ОРФ}} \subseteq K_{\text{ЧРФ}}.$$

Тезис Черча (для частично рекурсивных функций).

Класс алгоритмически вычислимых функций совпадает с классом всех частично рекурсивных функций. Принятие данного тезиса позволяет истолковывать доказательство, что некоторая функция не является частично рекурсивной, как доказательство отсутствия алгоритма вычисления ее значений.

Всякая функция, вычисляемая с помощью машины Тьюринга, является частично рекурсивной.

Всякая частично рекурсивная функция вычислима на машине Тьюринга.