Лекция 17
Рекурсивные функции

1. Вычислимые функции

Каждый **алгоритм задает функцию**, поскольку по набору исходных данных выдает результат применении алгоритма к этим данным

Совокупность тех элементов множества X, у которых есть соответствующие элементы в Y, называется **областью определения функции,** а совокупность элементов Y, называют **областью значений функции**

Если область определения функции из X в Y совпадает с множеством X, то функция называется **всюду определенной**

Функция y (x_1, x_2, ..., x_n) называется **вычислимой,** если существует алгоритм, позволяющий вычислить ее значение по известным значениям аргументов

Идея построения точного определения алгоритма, опирающегося на понятие вычислимой функции, состоит в том, что **любые дискретные данные можно закодировать натуральными числами** в некоторой системе счисления,

и тогда всякое их **преобразование сводится к последовательности вычислительных операций**,

а **результат обработки** также будет представлять собой **целое число.**

Какие функции могут быть вычислимыми?

Как описать такие алгоритмически вычислимые функции?

Исследование этих вопросов привело к созданию в 30х годах XX века ***теории рекурсивных функций*** (исторически первый подход к формализации понятия алгоритм)

2. Построение вычислимой функции

В теории РФ принят **конструктивный подход**: все множество исследуемых объектов (функций) строится из некоторого **базиса** с помощью простых ***операций***, вычислимость которых достаточно очевидна.

Операции над функциями принято называть операторами.

Все вычислимые функции можно построить на основе **трех элементарных функций** (базиса) путем применения к этим функциям **трех операторов**

2.1 Базисные функции

1) *Тождественное равенство нулю:*

$$O^n (x_1, x_2, \ldots, x_n) = 0$$

n-местная функция (функция от n аргументов), всегда возвращающая 0.

2) *Функция следования*

$$S^1(x) = x+1$$

Одноместная функция, сопоставляющая любому натуральному числу x непосредственно следующее за ним натуральное число x + 1

3) *Функция тождественного повтора одного из аргументов* (функция проекции):

$$I_m^n = (x_1, x_2, \ldots, x_n) = x_m$$

n-местная функция, сопоставляющая любому упорядоченному набору натуральных чисел число x_m из этого набора.

Пример1
Вычисление простейших функций

$$O^4 (2, 4, 6, 8) =$$

$$S^1 (2y+1) =$$

$$I_2^4 = (a, x, y, z) =$$

2.2. Операторы

1) *Оператор суперпозиции (подстановки)*

Оператором суперпозиции называется подстановка в функцию от n переменных n функций от m одних и тех же переменных. Суперпозиция дает новую функцию от n переменных.

Пусть m-местные функции

$$\boldsymbol{f_1}(x_1,x_2,\ldots,x_m), \boldsymbol{f_2}(x_1,x_2,\ldots,x_m), \ldots, \boldsymbol{f_n}(x_1,x_2,\ldots,x_m)$$

подставляются в n-местную функцию $\boldsymbol{g}(x_1,x_2,\ldots,x_n)$.

В результате получается n-местная функция

$$\boldsymbol{h}(y_1,y_2,\ldots,y_n) = \boldsymbol{g}(\boldsymbol{f_1}(x_1,x_2,\ldots,x_m), \boldsymbol{f_2}(x_1,x_2,\ldots,x_m), \ldots, \boldsymbol{f_n}(x_1,x_2,\ldots,x_m))$$

Говорят, что функция h получена из функций $g, f_1, \ldots, f_n$ *суперпозицией* (или подстановкой).

Обозначение: h = $S(g, f_1, \ldots, f_n)$

Если умеем вычислять функции $g, f_1, \ldots, f_n$,
то функция h также может быть вычислена.

Пример 2

Найти значение $S(S^1, O^1)$

$g(x) = S^1$, $f(x) = O^1 \to h(y) = g(f(x)) = S^1(O^1)$

Для этого значение простейшей функции O^1 должно быть подставлено в $S^1(x) = x + 1$.

Но $O^1(x) = 0$, следовательно,

$h(y) = S(S^1, O^1) = S^1(O^1) = 0+1 = 1.$

Пример 3

Найти значение $S(I^2_2, I^1_1, O^1)$

В этом случае конечная функция будет двухместной

следовательно $h(x_1,x_2) = I^2_2(I^1_1, 0^1) = 0^1 = 0$.

2) Оператор примитивной рекурсии

Оператор примитивной рекурсии определяет (n+1)-местную функцию **f** через n-местную функцию **g** и (n+2)-местную функцию **h** так:

$$f(x_1,x_2,\ldots,x_n,0) = g(x_1,x_2,\ldots,x_n),$$

$$f(x_1,x_2,\ldots,x_n, y+1) = h(x_1,x_2,\ldots,x_n,y,f(x_1,x_2,\ldots,x_n,y))$$

Приведенная пара равенств называется *схемой примитивной рекурсии*

Независимо от числа переменных в f рекурсия ведется только по одной переменной y. Остальные n переменных x1, x2, ..., xn на момент применения схемы зафиксированы и играют роль параметров.

При $y=0$ $\quad f(x_1,\ldots, x_n,0) = g(x_1,\ldots, x_n),$

При $y=1$ $\quad f(x_1,\ldots, x_n,1) = h(x_1,\ldots,x_n, 0, f(x_1,\ldots,x_n, 0)),$

При $y=2$ $\quad f(x_1,\ldots, x_n,2) = h(x_1,\ldots,x_n, 1, f(x_1,\ldots,x_n, 1)),$

….

$$f(x_1,\ldots, x_n,y+1)= h(x_1,\ldots,x_n, y, f(x_1,\ldots,x_n,y))$$

Если умеем находить значения функций g и h, то значения функции $f(x_1,\ldots, x_n, y + 1)$ можно вычислять «механически», находя последовательно значения на предыдущих шагах.

Операцию примитивной рекурсии обозначают

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

Пример 4.

Пусть $g(x) = x$ – функция от 1 переменной ($n=1$),

h - функция от $n+2 = 3$ переменных

$h(x,y,z) = x+y+z$

Найти функцию f от 2х аргументов- результат применения оператора примитивной рекурсии к паре функций g и h

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

$$f(x, 1) = h(x, 0, f(x,0)) = x+0+x = 2x$$

$$f(x, 2) = h(x, 1, f(x,1)) = x+1+2x = 3x+1$$

$$f(x, 3) = h(x, 2, f(x,2)) = x+2+3x+1 = 4x+3$$

$$f(x, 4) = h(x, 3, f(x,3)) = x+3+4x+3 = 5x+6$$

$$f(x, 5) = h(x, 4, f(x,4)) = x+4+5x+6 = 6x+10$$

$$f(x, 6) = h(x, 5, f(x,5)) = x+5+6x+10 = 7x+15$$

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2},$$

15=5 + 4 + 3 + 2 + 1 =

= (y-1)* (y-1+1) / 2 =

= (y^2-y)/2

….

$$f(x, y) = h(x, y-1, f(x,y-1)) = x+(y+1)*x+(y^2-y)/2$$

Если нужно доказать примитивную рекурсивность некоторой функции, нужно ее представить через простейшие функции и/или через функции примитивная рекурсивность которых уже доказана.

Пусть заданы два множества X и Y. Если **некоторым** элементам X поставлены в соответствие однозначно определенные элементы Y, то говорят, что задана **частичная функция** из X в Y

Частичная функция $f(x_1, x_2, \ldots, x_n)$ называется **примитивно рекурсивной,** если ее можно получить конечным числом операций суперпозиции и примитивной рекурсии, исходя лишь из простейших функций S^1, O^n и I^n_m

Пример 5. Покажем, что функция константа является примитивно-рекурсивной

$$f(x) = m = S^1(\ldots(S^1(O(x)))\ldots) \quad m \text{ раз}$$

Пример 6. Покажем, что двухместная функция *f(x,y) = x + y* является примитивно-рекурсивной

Так как функция f является функцией 2х аргументов, то для использования операции примитивной рекурсии мы должны иметь функцию g , зависящую от 1 аргумента, и функцию h , зависящую от 3 аргументов. Определим эти функции.

При y=0 fсум(x, 0) = x+0= x = $g(x)$ = $I^1_2(x,y)$ - функция проекции
При y=1 fсум(x, 1) = x+1 = $h(x, y, z)$ =$S^1(z)$ - функция следования

Используя схему примитивной рекурсии получаем:

f сум $(x,0) = g(x) = x,$

f сум $(x, 1) = h(x,0, f(x,0)) = S^1(f$сум $(x,0)) = x + 1$

f сум $(x, 2) = h(x, 1, f$сум$(x,1)) = S^1(f$сум$(x,1)) = x + 2$

. . .

fсум$(x, y) = h(x, y-1, f$сум$(x, y - 1)) = S^1(f$сум $(x, y - 1)) = (x + y-1)+1= x+y$

Функция $f(x,y)$ образуется из простейших функций следования и проекции операцией примитивной рекурсии и, следовательно, она сама примитивно рекурсивна.

Пример 7. Покажем, что двухместная функция $f(x, y) = x*y$ является примитивно-рекурсивной

Для этого мы должны показать, что функция f можно получить из базовых функций или функций, частичная рекурсивность которых уже доказана, путем применения операторов суперпозиции и примитивной рекурсии

Определим функции g и h

При y=0 fумн(x, 0) = x*0= 0 = $\boldsymbol{g(x) = O^1(x)}$ - баз.функц. тожд. равен. нулю

При y=1 fумн(x, 1) = x*1 = x = $\boldsymbol{h(x,y,z)= h(x,0,0)=}$ x+**fумн(x, 0)** - прим. рек ф-я сложения

Используя схему примитивной рекурсии получаем:

$f_{умн}(x,0) = g(x) = 0,$

$f_{умн}(x,1) = h(x,0, f_{умн}(x,0)) = x + f_{умн}(x,0) = x + 0 = x$

$f_{умн}(x,2) = h(x,1, f_{умн}(x,1)) = x + f_{умн}(x,1) = x + x = 2*x$

$f_{умн}(x,3) = h(x,2, f_{умн}(x,2)) = x + f_{умн}(x,2) = x + 2x = 3*x$

. . .

$f_{умн}(x, y) = h(x, y-1, f_{умн}(x, y - 1)) = x + f_{умн}(x,y-1) = x + (y-1)*x =$

$= x + x*y - x = x*y$

Функция $f(x,y)=x*y$ образуется из простейшей функции тождественного равенства нулю и примитивно-рекурсивной функции сложения операцией примитивной рекурсии и, следовательно, она сама примитивно-рекурсивна.

Пример 7. Покажем, что двухместная функция $f(x,y) = x^y$ является примитивно-рекурсивной

Так как функция f является функцией 2х аргументов, то для использования операции примитивной рекурсии мы должны иметь функцию g , зависящую от 1 аргумента, и функцию h , зависящую от 3 аргументов.

Определим эти функции.

При y=0   fст(x, 0) = x^0= 1 = $g(x)$ =S1(O1(x)) - суперпозиция нуль-функции и функции следования

При y=1   fст(x, 1) = x^1 = x* x^0

При y=2   fст(x, 2) = x^2 = x*(x* x^0)

$h(x, y, z)$ =x*z — функция умножения

Используя схему примитивной рекурсии получаем:

$f\ \text{ст}\ (x,0) = g(x) = 1,$

$f\ \text{ст}\ (x,1) = h(x,0,\ f\ \text{ст}(x,0)) = x*\ f\ \text{ст}\ (x,0)) = x*1 = x^1$

$f\ \text{ст}(x,\ 2) = h(x,\ 1,\ f\ \text{ст}(x,1)) = x*\ f\ \text{ст}(x,1) = x*x^1 = x^2$

$f\ \text{ст}(x,\ 3) = h(x,\ 2,\ f\ \text{ст}(x,2)) = x*\ f\ \text{ст}(x,2) = x*x^2 = x^3$

. . .

$f\ \text{ст}(x,\ y) = h(x,\ y-1,\ f\ \text{ст}(x,\ y-1)) = x*f\ \text{ст}\ (x,\ y-1) =$

$= x*x^{(y-1)} = x^y$

Функция $f(x,y)=x^y$ образуется из суперпозиции простейших функции тождественного равенства нулю, следования и примитивно-рекурсивной функции умножения операцией примитивной рекурсии и, следовательно, она сама примитивно-рекурсивна.

2°. Сложение $x+y$ является рекурсивной функцией. Оно задается схемой

$$\begin{cases} x+0=x=I_1^2(x, y), \\ x+(y+1)=(x+y)+1=s(x+y). \end{cases}$$

Здесь $g(x)=I_1^2(x, y)$, $h(x, y, z)=s(z)$.

3°. Умножение $x \cdot y$ является рекурсивной функцией. Оно задается схемой

$$\begin{cases} x \cdot 0 = 0 = 0(x), \\ x \cdot (y+1) = x \cdot y + x. \end{cases}$$

Здесь $g(x) = 0(x)$, $h(x, y, z) = x + z$ (сложение рекурсивно).

Таким образом, из простейших функций с помощью операторов суперпозиции и примитивной рекурсии можно получить множество функций, включая основные функции арифметики, алгебры и анализа.

Т.е. эти функции имеют примитивно-рекурсивное описание, которое однозначно определяет процедуру их вычисления. Следовательно они являются вычислимыми функциями

Однако, не все вычислимые функции можно описать как примитивно-рекурсивные (пример ф-я Аккермана)

т. е. понятие примитивно-рекурсивная функция не является точным формальным аналогом неформального понятия алгоритм, им является понятие **частично-рекурсивная функция**

Функция $f(x_1,x_2,\ldots,x_n)$ называется **частично рекурсивной,** если ее можно получить с помощью конечного числа операторов суперпозиции, примитивной рекурсии и

μ- оператора, исходя лишь из простейших функций S^1, O^n и I^m_n

3) μ-оператор (оператор минимизации для функции n аргументов)

Пусть задана функция $f(x_1,x_2,\dots,x_{n-1}, y)$.
Зафиксируем значения $x_1,x_2,\dots,x_{n-1}$, и выясним,
при каких y значение $f(\mathbf{x_1},\mathbf{x_2},\dots,\mathbf{x_{n-1}},y) = 0$.

Можно найти *наименьшее* из тех значений y,
при которых $f(\mathbf{x_1},\mathbf{x_2},\dots,\mathbf{x_{n-1}},y) = 0$.

Примем обозначение:

$$F(x_1,x_2,\dots,x_{n-1}) = \mu_y[f(x_1,x_2,\dots,x_{n-1},y) = 0]$$

(читается: *«наименьшее y такое, что* $f(x_1,x_2,\dots,x_{n-1},y) = 0$»,
а μ_y называют μ *-оператором или оператором минимизации)*.

Оператор минимизации “следит”, при каком значении выбранного аргумента наблюдаемая им функция впервые опустится до нуля. Это значение выбранного аргумента и будет значением оператора минимизации.

Например, для функции $x-y$, при $x = 5$, значение оператора минимизации также будет равно 5, поскольку двигаясь в значениях игрека от нуля получим нулевое значение функции именно при игрек равном 5.

Работа μ-оператора

Рассмотрим $F(x_1,x_2,\ldots,x_{n-1}) = \mu_y[f(x_1,x_2,\ldots,x_{n-1},y) = 0]$,
где y - выделенная переменная.

Работу μ-оператора можно описать следующим образом;

Выделяется переменная (здесь - y). Затем фиксируется значение остальных переменных ($x_1, \ldots, x_{n-1}$).

Значение y последовательно увеличивается, начиная с нуля.

Значением μ-оператора будет значение y, при котором функция впервые обратилась в ноль.

Значение μ-оператора считается неопределенным, если функция вообще не принимает значения ноль, либо она принимает отрицательное значение до того как примет значение ноль.

Работа μ-оператора

Для вычисления функции F:

Вычисляем $f(x_1,x_2,\dots,x_n,0)$; если значение равно нулю, то полагаем $F(x_1,x_2,\dots,x_n) = 0$. Если $f(x_1,x_2,\dots,x_n,0) \neq 0$, то переходим к следующему шагу.

Вычисляем $f(x_1,x_2,\dots,x_n,1)$; если значение равно нулю, то полагаем $F(x_1,x_2,\dots,x_n) = 1$. Если $f(x_1,x_2,\dots,x_n,1) \neq 0$, то переходим к следующему шагу и т.д.

Если окажется, что для всех y функция $f(x_1,x_2,\dots,x_n,0) \neq 0$, то функция $F(x_1,x_2,\dots,x_n)$ считается неопределенной.

Пример 8. работа μ-оператора

Пусть функция $g(x, y) = x - y + 3$.

Зафиксируем $x = 1$

$F(x,y) = \mu_y[g(1, y) = 0] = \mu_y[1 - y + 3= 0] = 4$

так как $1 - 4 + 3 = 0$.

Доказано, что:

множество частично-рекурсивных функций совпадает с множеством *вычислимых функций - алгоритмически разрешимых задач*.

Рассмотрим как выполняются основные требования к алгоритмам для алгоритмической модели „частично-рекурсивные функции“

Детерминированность определяется полной определенностью в вычислении базисных функций и полной заданностью в действиях операторов.

Там же показана **элементарность** каждого шага и **дискретность** вычислений

Результатом является значение частично-рекурсивной функции, вычисляемой в процессе применения операторов

Возможность выбора в качестве аргумента любого натурального числа обеспечивает **массовость** частично-рекурсивной функции

Таким образом, понятие ЧРФ является исчерпывающей формализацией понятия алгоритма.

Это выражено в виде **тезиса Чёрча**:

всякий алгоритм может быть реализован частично-рекурсивной функцией.

Утверждение о несуществовании частично-рекурсивной функции эквивалентно несуществованию алгоритма решения соответствующей задачи.

Вопросы к лекции

1. Как понятие алгоритма связано с понятием вычислимой функции?
2. Перечислите простейшие функции и элементарные операторы.
3. Почему примитивно-рекурсивную функцию нельзя считать точным формальным определением понятия алгоритм?
4. Как можно получить частично-рекурсивную функцию?
5. Сформулируйте определение алгоритма с позиций теории ЧРФ?

Семинар

Задание 1. Применение оператора суперпозиции

Даны 3 функции от двух переменных:

$f_1(x_1, x_2) = 2\ x_1+3x_2$

$f_2(x_1, x_2) = x_{1*}x_2$

$f_3(x_1, x_2) = x_1 * \sqrt{x_2}$

и функция

$g(z_1, z_2, z_3) = z_1+z_2 * z_3$

Найти функцию h - суперпозицию функций f в функцию g

$h = S(g,f)$

Задание 2 (самостоятельно)

Даны три одноместные функции:

$f1(x) = O^1(x)$

$f2(x) = O^1(x)$

$f3(x) = S^1(10)$

и трехместная функция $g(y1, y2, y3) =$ $I_3^3(y1, y2, y3)$

$h = S\,(g, f_1, f_2, f_3)$ - ?

Задание 3
Применение оператора примитивной рекурсии

Пусть

$g(x) = 0$ – функция от 1 переменной

$h(x,y,z) = x+z$ – функция от 3 переменных

Найти функцию $f(x, y)$ — функцию от 2 переменных, путем применения оператора примитивной рекурсии к функциям g и h

$$h = R(g, h) \text{ - ?}$$

Задание 4 (самостоятельно)

Пусть

g(x1,x2) = x1*x2 – функция от 2 аргументов

h(x1,x2,x3,x4) = x1*x2+x3*x4 – функция от 4 аргументов

Построить выражения для функции f (x1, x2, x3) от 3 аргументов, путем применения оператора примитивной рекурсии к функциям g и h для первых 6 шагов рекурсии

h = R(g, h) - ?

Задание 5
Применение оператора минимизации

Рассмотрим функцию *F(x,y) = x - y,* которая может быть получена с помощью применения оператора минимизации к функции f(x,y,z) = x-y-z:

$F(x,y) = \mu_z$ [x-y-z=0]

Вычислим, например, F*(7,2),* т.е. значение функции при *y = 2* и *x = 7.*

При z=5, μ_z [x-y-z=0]

Таким образом, найдено значение функции F*(7,2)* = 5.