

Теория алгоритмов

Вычислимые функции

Алгоритмы: определение и основные свойства

Понятие алгоритма является одним из основных понятий современной математики. Само слово “алгоритм” является производным от имени среднеазиатского ученого Аль Хорезми, уроженца Хивы, жившего в IX веке нашей эры.

Чтобы иметь возможность более уверенно решать алгоритмические задачи, возникающие в различных разделах теоретической и прикладной математики, необходимо иметь достаточно развитую теорию алгоритмов.

Алгоритм – это точное предписание о выполнении в некотором порядке системы операций, определяющих процесс перехода от исходных данных к искомому результату для решения задачи данного типа.

Это понятие относится к исходным математическим понятиям, которые не могут быть определены через другие, более простые понятия. Иногда такое или подобное определение называют ***интуитивным***, т.е. понятным из опыта.

Предписание считается алгоритмом, если оно обладает следующими *свойствами*:

- **определенностью**, то есть общепонятностью и точностью, не оставляющими место произволу,
- **массовостью**, то есть возможностью исходить из меняющихся в известных пределах значений исходных данных,
- **результативностью**, то есть направленностью на получение искомого результата,
- **элементарностью шагов**, на которые разбивается последовательность операций.

Каждый алгоритм, в общем случае, должен задаваться следующими параметрами:

- множеством допустимых исходных данных,
- начальным состоянием,
- множеством допустимых промежуточных состояний,
- правилами перехода из одного состояния в другое,
- множеством конечных результатов,
- конечным состоянием.

Перечисленных свойств вполне достаточно, чтобы можно было определить, является ли данное конкретное предписание алгоритмом.

Необходимость уточнения понятия алгоритма

Длительное время интуитивного понятия алгоритма было вполне достаточно для задач математики и логики. Пусть имеется некоторая алгоритмическая проблема, т.е. необходимо найти алгоритм для решения определенного класса задач. Если такой алгоритм найден, то предложенный способ для решения задачи признается всеми как алгоритм, исходя из их собственного интуитивного представления об алгоритме. Поэтому не требуется точного определения понятия алгоритма.

Однако, ситуация становится противоположной, если у нас после длительных, безуспешных попыток нахождения алгоритма возникает гипотеза, что такого алгоритма не существует.

Чтобы иметь возможность доказывать несуществование алгоритма, мы должны математически точно определить объект, существование которого будет опровергаться.

Для доказательства несуществования алгоритма необходимо располагать точным определением понятия алгоритма.

Формализации Черча, Тьюринга и Маркова.

В двадцатых годах XX века задача точного определения понятия алгоритма стала одной из центральных математических проблем. Она заключалась в попытке дать строгое математическое определение алгоритма, которое соответствовало бы имевшимся в то время интуитивным представлениям об алгоритме. Решение данной проблемы было получено в первой половине XX века. Мы рассмотрим следующие три варианта определения алгоритма.

1) С точки зрения А.Черча и С. Клини, в алгоритмическом процессе вычисляется значение некоторой функции $f(x_1, x_2, \dots, x_n)$, определенной на множестве натуральных чисел. Строгое определение алгоритма, предложенное Черчем и Клини, основано на понятии частично рекурсивной функции. Они предложили отождествить интуитивное понятие «алгоритм» со строгим математическим понятием «частично рекурсивная функция».

- 2) С точки зрения английского математика А.Тьюринга алгоритмический процесс — это работа некоторой воображаемой вычислительной машины — машины Тьюринга. Он предложил отождествить интуитивное понятие «алгоритм» с точным понятием «машина Тьюринга».**
- 3) Третий вариант определения алгоритма предложил в 1947 г. российский математик А.А.Марков. С его точки зрения, алгоритмический процесс – это переработка слов некоторого алфавита с помощью точно описанных правил переработки. Вместо интуитивного понятия «алгоритм» А.А.Марков вводит строгое понятие «нормальный алгорифм».**
- 4) В качестве вычислительной машины будет также рассмотрена машина с неограниченными регистрами (МНР) , предложенные Шепердсоном и Стерджисом в 1963 г. Это абстрактная машина, более сходная с реальным компьютером по сравнению с машиной Тьюринга.**

Каждый из этих вариантов определения понятия алгоритма фиксирует свой класс K рассматриваемых объектов, свои правила переработки, окончания и опознания результата. Однако, все эти варианта определения понятия алгоритма равносильны. Это означает, что при подходящей кодировке объекты из K можно трактовать в виде объектов, с которыми работают при другом определении алгоритма. То же самое с правилами переработки, правилами окончания и опознания результата.

Для всех определений и утверждений при одном определении алгоритма возникают аналогичные определения и утверждения при другом определении. Поэтому мы имеем взаимную моделируемость этих понятий.

Это указывает на то, что эти различные определения отражают в разных формах понятие алгоритма.

Счетные множества.

Напомним некоторые сведения о счетных множествах. Множество A называется равномошным множеству B , если существует биективное (взаимно однозначное) отображение множества A во множество B . Если A равномошно B , то B равномошно A .

Множество A называется счетным множеством, если оно равномошно множеству натуральных чисел \mathbb{N} . В этом случае существует биекция $f: \mathbb{N} \rightarrow A$ и элемент a с условием $f(n) = a$ можно обозначить через a_n . Получим запись $A = \{a_0, a_1, a_2, \dots\}$, где $a_i \neq a_j$ при $i \neq j$. Обратно, из такой записи следует счетность множества A . Получили следующее утверждение.

Определение. Множество A счетно тогда и только тогда, когда его элементы можно представить в виде бесконечной последовательности без повторений $A = \{a_0, a_1, a_2, \dots\}$, где $a_i \neq a_j$ при $i \neq j$.

Пример. Множество целых чисел \mathbb{Z} счетно, так как целые числа можно расположить в виде последовательности $0, 1, -1, 2, -2, 3, -3, \dots$.

Отметим следующие известные свойства счетных множеств.

- 1) Подмножество счетного множества конечно или счетно.
- 2) Объединение конечного или счетного числа конечных или счетных множеств конечно или счетно.
- 3) Множество действительных чисел не является счетным множеством.
- 4) Множество M , состоящее из всех строк $\{x_1, x_2, \dots, x_n\}$ произвольной длины $n \geq 0$, где $x_i \in N$, является счетным множеством.
- 5) Пусть A — конечный алфавит и X — множество всех слов в алфавите A . Тогда множество X является счетным множеством.

Пусть M — бесконечное множество конструктивных объектов, являющихся множеством входных объектов некоторого алгоритма. Множество M можно рассматривать как множество слов в некотором алфавите. По пункту 5) множество M является счетным множеством.

Арифметическая функция – функция, определенная на множестве натуральных чисел и принимающая значения из множества натуральных чисел.

Теорема. *Множество арифметических функций n -переменных несчетно.*

Доказательство

Для доказательства несчетности множества достаточно доказать несчетность какого-нибудь его подмножества.

Рассмотрим арифметические функции одной переменной вида $f_i(x)$.

Эти арифметические функции одной переменной образуют подмножество множества арифметических функций n переменных.

Предположим противное. Пусть арифметических функций одной переменной счетное множество, т.е. их можно перечислить.

Тогда их можно расположить в виде бесконечной последовательности $f_0(x), f_1(x), f_2(x), \dots, f_n(x), \dots$

Построим новую функцию $g(x) = f_x(x) + 1$.

Это так называемая диагональная функция, например:

$g(0) = f_0(0) + 1, g(1) = f_1(1) + 1, g(2) = f_2(2) + 1, \dots$

$g(x)$ отлична от всех перечисленных функций, т.к. от каждой из функций она отличается хотя бы в одной точке. Например, от функции $f_0(x)$ функция $g(x)$ отличается в точке $x=0$, от функции $f_1(x)$ функция $g(x)$ отличается в точке $x=1$ и т.д.

Однако по построению $g(x)$ принадлежит множеству арифметических функций одной переменной, значит должна быть в списке перечисления $f_i(x)$, т.е. совпадать с одной из перечисленных функций. Получили противоречие, следовательно, исходное предположение неверно, и функций одной переменной несчетное множество.

Поскольку множество арифметических функций одной переменной является подмножеством множества арифметических функций n переменных, то значит множество арифметических функций n переменных несчетно, Q.E.D.

Арифметическая функция называется вычислимой, если существует алгоритм для ее вычисления в каждой точке.

Алгоритмы и функции.

Вариант строгого определения алгоритма, предложенный Черчем и Клини, рассматривает алгоритмический процесс как процесс вычисления значений некоторой функции f , определенной на множестве \mathbb{N} .

Примитивно рекурсивные функции

Алгоритмический процесс можно рассматривать как процесс вычисления значений некоторой функции f . Такую функцию f называют вычислимой функцией. Интуитивное понятие вычислимой функции мы заменим на точное понятие частично рекурсивной функции. В результате получим строгое определение алгоритма.

Исходные функции. Рассмотрим некоторый набор простейших функций, вычислимость которых очевидна.

Функция следования.

Рассмотрим функцию $s(x)$, заданную правилом

$$s(x) = x + 1 \text{ для всех } x \in \mathbb{N}.$$

Эта функция очевидно вычислима, т.к. алгоритм ее вычисления состоит из простейшего действия «добавления к x палочкам еще одной палочки».

Функция $s(x)$ называется функцией следования.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Оператор суперпозиции S.

Рассмотрим действие, которое назовем оператором суперпозиции (регулярной суперпозиции). С помощью этого действия из некоторых функций h, g_1, g_2, \dots, g_m создается новая функция f .

Пусть дана функция h от m переменных и заданы m функций g_1, g_2, \dots, g_m от n переменных. При этом $m, n \geq 0$.

Новую функцию $f(x_1, x_2, \dots, x_n)$ определим по следующему правилу:
 $f(x_1, x_2, \dots, x_n) = h(g_1(x_1, x_2, \dots, x_n), g_2(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n))$.

Функция f является частичной функцией от n переменных. Ее значение $f(x_1, x_2, \dots, x_n)$ определено тогда и только тогда, когда определены все выражения в правой части равенства. Если функции h и g_1, g_2, \dots, g_m вычислимы, то и функция f вычислима. Алгоритм ее вычисления описывается правой частью равенства. Функция f получена оператором суперпозиции из функций h и g_1, g_2, \dots, g_m , и записывается как

$$f = S(h, g_1, g_2, \dots, g_m).$$

Пусть $m = 1, n = 1$ в равенстве для f и функция $f(x)$ получена оператором суперпозиции из функций $h(x)$ и $g(x)$. Тогда имеем запись

$$f(x) = h(g(x)) \text{ для всех } x \in N.$$

Оператор примитивной рекурсии R .

Рассмотрим действие, которое назовем оператором примитивной рекурсии. С помощью оператора примитивной рекурсии мы конструируем функцию f от $n + 1$ переменных из некоторых частичных функций g и h . При этом функция g имеет $n \geq 0$ переменных, а функция h имеет $n + 2$ переменных. Значения новой функции f вычисляем по двум правилам:

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

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

Слово рекурсия (recursio на латинском языке — возвращаюсь) означает вычисление значения $f(x_1, x_2, \dots, x_n, y + 1)$ с использованием $f(x_1, x_2, \dots, x_n, y)$ (возвращением к ранее вычисленному значению). Таким образом, получаем последовательное вычисление значений функции f :

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

$$f(x_1, x_2, \dots, x_n, 1) = h(x_1, x_2, \dots, x_n, 0, f(x_1, x_2, \dots, x_n, 0)).$$

...

$$f(x_1, x_2, \dots, x_n, m + 1) = h(x_1, x_2, \dots, x_n, m, f(x_1, x_2, \dots, x_n, m)).$$

Если все значения в правых частях равенств существуют, то получим значение функции $f(x_1, x_2, \dots, x_n, m + 1)$.

Если какое-то значение не определено, то $f(x_1, x_2, \dots, x_n, m + 1)$ не существует. Поэтому в общем случае мы получаем частичную функцию $f(x_1, x_2, \dots, x_{n+1})$ от $n + 1$ переменной.

В случае $n = 0$ функция g имеет 0 переменных и поэтому отождествляется с некоторым числом a . Тогда равенство имеет вид

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

Если функция f получена оператором примитивной рекурсии из функций g и h , то записываем

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

Как и в случае оператора суперпозиции, вычислимость исходных функций g и h влечет вычислимость построенной из них функции f .

Функция f называется **примитивно-рекурсивной**, если ее можно получить конечным числом операций подстановки и примитивной рекурсии, исходя лишь из простейших функций S , O , Γ_m^n .

Операции подстановки и примитивной рекурсии, будучи применены к всюду определенным функциям, дают в результате снова всюду определенные функции. В частности всюду определенными будут все примитивно-рекурсивные функции.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Пример 3. Постоянная унарная функция $f(x) = a$ примитивно рекурсивна.

Доказательство. Имеем запись $f(x) = s(s(\dots s(o(x))\dots)) = a$, где функции $s(x)$ и $o(x)$ — простейшие функции. Применяя оператор суперпозиции несколько раз, получим что функция $f(x) = a$ примитивно рекурсивна.

Пример 4. Нульарная функция примитивно рекурсивна.

Доказательство. Пусть f — нульарная функция, значения которой равны a .

Нульарная нулевая функция $g = o^0$ имеет 0 переменных и отождествляется с числом $0 \in \mathbb{N}$. Она примитивно рекурсивна, т.к. является простейшей функцией. В следующем равенстве функция следования s повторена a раз, т.е. a раз применен оператор суперпозиции

$$s(s(\dots s(g))) = a.$$

Функция в левой части примитивно рекурсивна. Она имеет 0 переменных

и тождественно равна a . Эта функция является нульарной функцией f , примитивную рекурсивность которой нам требовалось установить, ч.т.д.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Утверждение. Функция сложения $f(x, y) = x + y$ примитивно рекурсивна.

Доказательство. Покажем, что функция сложения $f(x, y)$ может быть получена с помощью оператора примитивной рекурсии из двух функций $g(x)$ и $h(x, y, z)$, для которых примитивная рекурсивность уже установлена.

Способ для подбора функций g и h в общем случае состоит в выполнении двух действий.

1) Равенство $f(x_1, x_2, \dots, x_n, 0) = g(x_1, x_2, \dots, x_n)$, и получим функцию g . В нашем случае $n = 1$ и $f(x, y) = x + y$. Имеем $g(x) = f(x, 0) = x + 0 = x$.

Получили искомую функцию $g(x) = x$. То, что функция $g(x)$ примитивно рекурсивна, уже известно по предложению 2.

2) Равенство $f(x_1, x_2, \dots, x_n, y + 1) = h(x_1, x_2, \dots, x_n, y, f(x_1, x_2, \dots, x_n, y))$, и подберем функцию $h(x, y, z)$. Для функции $f(x, y) = x + y$ получим равенства $f(x, y+1) = x + (y+1) = (x+y)+1 = s(x+y) = s(f(x, y)) = h(x, y, f(x, y))$.

При этом $h(x, y, z) = s(z)$. Функция $h(x, y, z)$ получена из примитивно рекурсивной функции $s(z)$ введением двух фиктивных переменных x и y . Поэтому функция $h(x, y, z)$ примитивно рекурсивна. Получили, что функция $f(x, y) = x + y$ примитивно рекурсивна, ч.т.д.

Утверждение. Функция умножения $f(x, y) = x \cdot y$ примитивно рекурсивна.

Доказательство. Как и в предыдущем предложении подберем функции $g(x)$ и $h(x, y, z)$, применяя правила рекурсии.

Имеем $g(x) = f(x, 0) = x \cdot 0 = 0$. Поэтому $g(x)$ — нулевая функция $o(x)$, которая примитивно рекурсивна.

Покажем, что в качестве искомой функции $h(x, y, z)$ можно взять функцию $h(x, y, z) = x + z$, которая примитивно рекурсивна, т.к. получена из функции сложения введением фиктивной переменной y . Имеем

$$f(x, y + 1) = x \cdot (y + 1) = x + x \cdot y = x + z \text{ при } z = x \cdot y = f(x, y).$$

Тогда $f(x, y + 1) = h(x, y, f(x, y))$. Из примитивной рекурсивности функций $g(x)$ и $h(x, y, z)$ заключаем, что функция $f(x, y) = x \cdot y$ примитивно рекурсивна

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Определение. Арифметическая функция f называется примитивно рекурсивной функцией (ПРФ), если существует ПРО, последней функцией которого f является.

Пример. Функции $x + y$ и $x \cdot y$ примитивно рекурсивны.

Отметим следующие свойства ПРО.

1. Каждый начальный отрезок ПРО есть ПРО.
2. Каждая функция в ПРО примитивно рекурсивна.
3. Если в ПРО в любом его месте между двумя соседними функциями вставить ПРФ, то получим снова ПРО.
4. Если в ПРО в любом его месте между двумя соседними функциями вставить какое-либо другое ПРО, то получим снова ПРО.

Замечание. Всякая ПРФ всюду определена, ибо исходные функции всюду определены, и операции подстановки и примитивной рекурсии сохраняют всюду определенность функций.

Определение. Пусть N есть конечная совокупность функций. Прimitивно рекурсивное описание относительно совокупности функций N есть конечная последовательность функций, каждая из которых есть либо исходная функция, либо функция из совокупности N , либо получена из предыдущих функций последовательности с помощью операции подстановки или примитивной рекурсии.

Определение. Функция f примитивно рекурсивна относительно совокупности функций N , если существует ПРО относительно совокупности N , последней функцией которого функция f является.

Свойства ПРО относительно совокупности

функций H .

1. Если функция f примитивно рекурсивна относительно совокупности функций H , а функции из H входят в совокупность функций M , то функция f примитивно рекурсивна относительно совокупности M .

2. Если функция f примитивно рекурсивна относительно совокупности функций H и совокупность функций G из H состоит только из ПРФ, то функция f примитивно рекурсивна относительно совокупности $H - G$.

3. Если функция f примитивно рекурсивна относительно совокупности функций H и каждая функция из H примитивно рекурсивна относительно совокупности функций G , то функция f примитивно рекурсивна относительно совокупности G .

4. Прimitивно рекурсивная функция примитивно рекурсивна относительно любой совокупности функций.

5. Если функция f примитивно рекурсивна относительно совокупности функций H и G есть совокупность ПРФ, то функция f примитивно рекурсивна относительно совокупности $H \cup G$.

К свойствам примитивно рекурсивного описания относительно совокупности функций следует добавить свойства ПРО.

Функции, представимые термами

Формальные символы

1. f_1, f_2, \dots, f_3 - символы для арифметических функций.
2. $0, 1, 2, \dots$ - символы натуральных чисел.
3. x, y, z, \dots - символы переменных.
4. $(,)$ - скобки левая и правая и запятая.

Термы

1. Символ натурального числа есть терм ранга 1.
2. Символ переменной есть терм ранга 1.
3. Если f_k есть функциональный символ, а t_1, \dots, t_n - термы, максимальный ранг которых равен k , то выражение $f_k(t_1, \dots, t_n)$ есть терм ранга $k+1$.

Пример, $x; 0; 1; 2; 10; 17; f_1(x,1); f_2(17,x,10); f_1(f_1(x,2), f_2(17,x,10))$.

Замечание. Содержательно терм есть некоторая подстановка (суперпозиция) функций. Иногда ранг терма называют глубиной построения терма.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Шаг индукции. Покажем, что функция $f_k(t_1, \dots, t_{nk})$, представимая термом $f_k(t_1, \dots, t_{nk})$ ранга r , примитивно рекурсивна относительно совокупности функций F .

Так как термы t_1, \dots, t_{nk} имеют ранг меньше r , то представимые ими функции t_1, \dots, t_{nk} примитивно рекурсивны относительно совокупности F по предположению индукции. Подходящими подстановками в них функций выбора и констант мы придем к функциям одинаковой местности u_1, \dots, u_{nk} . Тогда функция $g = f_k\{t_1, \dots, t_{nk}\} = f_k(u_1, \dots, u_{nk})$. Так как функции u_1, \dots, u_{nk} примитивно рекурсивны относительно F , то функция g примитивно рекурсивна относительно F , ибо получена подстановкой из функций, примитивно рекурсивных относительно F .

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Рекурсивные определения знакомы математику.

Например, функция f , определяемая соотношениями

$$f(0) = 1,$$

$$f(1) = 1,$$

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

дает ряд Фибоначчи: 1, 1, 2, 3, 5, 8, 13, ... (Исследование разностных уравнений связано с проблемой перехода от рекурсивного определения к алгебраическому выражению. Ряд

$$f(x) = \frac{\sqrt{5}}{5} \left(\frac{1 + \sqrt{5}}{2} \right)^{x+1} - \frac{\sqrt{5}}{5} \left(\frac{1 - \sqrt{5}}{2} \right)^{x+1} :$$

Примитивнорекурсивные функции — это пример широкого и интересного класса всюду определенных функций, который можно получить подобной формализацией.

Мы рассматриваем точное понятие примитивнорекурсивной функции как частный случай неформального понятия *всюду определенной функции, вычислимой алгоритмом*.

Насколько богат класс примитивнорекурсивных функций? Быть может, он включает все алгоритмы и, следовательно, дает точный формальный аналог неформального понятия *всюду определенной функции, вычислимой алгоритмом*? Хотя на первый взгляд правила для построения примитивнорекурсивных функций кажутся ограниченными, можно представить убедительные свидетельства в пользу этого предположения. Можно показать, что практически все всюду определенные функции обычной математики примитивнорекурсивны.

К сожалению, можно построить и всюду определенные функции с очевидными алгоритмами, которые нб являются примитивно-рекурсивными. Одна из них — это функция Аккермана, т. е. функция трех переменных, такая, что:

$$f(0, 0, y) = y,$$

$$f(0, x+1, y) = f(0, x, y)+1,$$

$$f(1, 0, y) = 0,$$

$$f(s+2, 0, y) = 1,$$

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

Для этой функции не существует примитивно-рекурсивной схемы

Так как функция Аккермана — это всюду определенная функция, вычисляемая алгоритмом, и так как она не примитивно-рекурсивна, то понятие примитивно-рекурсивной функции не является точным формальным аналогом неформального понятия всюду определенной алгоритмической функции.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.

Нулевая функция.

Пусть $n \geq 0$. Рассмотрим n -арную функцию $o^n(x_1, x_2, \dots, x_n)$, заданную правилом

$$o^n(x_1, x_2, \dots, x_n) = 0 \text{ для всех } x_1, x_2, \dots, x_n \in \mathbb{N}.$$

Данная функция называется нулевой функцией. Нулевая функция, как и функция следования, очевидно вычислима. Нулевую функцию при $n = 1$ обозначаем через $o(x)$. Поэтому

$$o(x) = 0 \text{ для всех } x \in \mathbb{N}.$$

Нулевая функция при $n = 0$ обозначается через o^0 и отождествляется с числом 0.

Функция проектирования (выбора).

Пусть $n \geq 1$ и $1 \leq m \leq n$. Зададим функцию проектирования $I_m^n(x_1, x_2, \dots, x_n)$ правилом

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

Данная функция строке аргументов x_1, x_2, \dots, x_n сопоставляет ее компоненту x_m с номером m . Вычислимость функции проектирования обеспечивается нашей способностью найти в строке x_1, x_2, \dots, x_n место с номером m и указать число на этом месте.