# Примитивно-рекурсивные функции

## Практическое занятие 1

## Теория рекурсивных функций

В теории рекурсивных функций, как и вообще в теории алгоритмов, принят конструктивный (финитный) подход, основной чертой которого является то, что все множество исследуемых объектов (в данном случае функций) строится из конечного числа исходных объектов — базиса — с помощью простых операций, эффективная вычислимость которых достаточно очевидна. Операции над функциями называют операторами.

Для простоты будем рассматривать только числовые функции, т.е. функции, аргументы и значения которых принадлежат множеству натуральных чисел (в теории рекурсивных функций полагают ). Иначе говоря, числовой -местной функцией называется функция, определенная на некотором подмножестве с натуральными значениями. Если область определения совпадает с множеством , то говорят, что функция всюду определена, в противном случае — частично определена.

**Частичными** числовыми функциями , где (), называют функции, определенные не на всех наборах .

## Простейшие числовые функции и операторы

* **Нуль-функция**: ;
* **Функция следования**: ;
* **Функция проекции**: ().

### Операторы

* **Оператор суперпозиции**
* **Оператор примитивной рекурсии**
* **Оператор минимизации**

## Оператор суперпозиции

Оператором суперпозиции называется подстановка в функцию от переменных функций, каждая из которых зависит от одних и тех же переменных. Суперпозиция дает новую функцию уже от переменных. Например, для функций , , , , их суперпозиция дает новую функцию :

В этом случае говорят, что -местная функция получена с помощью оператора суперпозиции из -местной функции и -местных функций , , , , если

## Оператор примитивной рекурсии

Оператор примитивной рекурсии определяет -местную функцию через -местную функцию и -местную функцию следующим образом:

Пара равенств называется схемой примитивной рекурсии.

Тот факт, что функция определена схемой, выражается равенством

Эта схема определяет рекурсивно не только через другие функции, но и через значения в предшествующих точках: значение в точке зависит от значения в точке .

Для вычисления понадобится вычислений по указанной схеме для .

Существенным в операторе примитивной рекурсии является то, что независимо от числа переменных в , рекурсия ведется только по одной переменной , а остальные переменных на момент применения схемы зафиксированы и играют роль параметров.

В случае, когда , т.е. определяемая функция является одноместной, схема принимает более простой вид:

где — константа.

Функция называется **примитивно-рекурсивной**, если она может быть получена из нуль-функции , функции следования и функции проекции с помощью конечного числа применений операторов суперпозиции и примитивной рекурсии.

Этому определению можно придать более формальный индуктивный вид:

1. Функции , и для всех натуральных , где , являются примитивно-рекурсивными.
2. Если , $, , — примитивно-рекурсивные функции, то — примитивно-рекурсивные функции для любых натуральных .
3. Если и — примитивно-рекурсивные функции, то — примитивно-рекурсивная функция.
4. Других примитивно-рекурсивных функций нет.

## Оператор минимизации

Пусть задана некоторая функция . Зафиксируем значение и выясним, при каком функция .

Более сложной задачей является отыскание для данной функции и фиксированного наименьшего из тех значений , при которых функция . Так как результат решения задачи зависит от , то наименьшее значение , при котором функция , есть функция . Принято обозначение

которое читается как: «наименьшее такое, что ».

Аналогично определяется функция многих переменных:

Переход от функции к функции принято называть применением **-оператора**.

### Алгоритм вычисления функции

1. Вычислим . Если это значение равно нулю, то полагаем . Если , то переходим к следующему шагу.
2. Вычислим . Если , то полагаем . Если же , то переходим к следующему шагу. И так далее.

Если окажется, что для всех функция , то функцию в этом случае считают неопределенной. Но возможно, что существует такое , что и, значит, есть и наименьшее , при котором , и в то же время может случиться, что при некотором () значение функции не определено. Очевидно, что в этом случае процесс вычисления наименьшего , при котором , не дойдет до . И здесь функцию считают неопределенной.

## Методология составления схем примитивной рекурсии

Для составления схемы примитивной рекурсии первоначально необходимо определить вид нужной схемы. Для этого следует ответить на вопрос: «От скольких переменных зависит данная функция?» Допустим, переменных . Значит, используется схема рекурсии с параметрами.

Следующие шаги:

1. Если не указано, по какой переменной составлять схему, то следует выбрать наиболее простую переменную (наименьшее количество операций с этой переменной или наиболее простые операции). Если при ответе на вопрос: «От скольких переменных зависит данная функция?» получено, что такая переменная одна единственная, значит, используется схема рекурсии без параметров.
2. Определить, чему равно значение функции в точке ноль? (верхний индекс в функциях и операциях для будет равен ).
3. Определить, чему равно значение функции в следующей точке (переменная, по которой берется рекурсия, плюс единица). Постарайтесь преобразовать получившееся значение к самой функции, верхний индекс в функциях и операциях для любого значения будет равен .
4. Записать функцию через операцию примитивной рекурсии.

## Примеры

# Доказать примитивную рекурсивность суммы

**Решение.**

Таким образом,

где .

Действительно, имеем:

# Доказать примитивную рекурсивность произведения

**Решение.**

Или , где .

Выполняя действия по полученной схеме, получаем:

# Доказать примитивную рекурсивность возведения в степень

**Решение.**

Действительно, выполняя вычисления по полученной схеме, получаем:

# Задания

1. Доказать примитивную рекурсивность факториала .
2. Доказать примитивную рекурсивность псевдоразности
3. Доказать примитивную рекурсивность знаковой функции
4. Доказать примитивную рекурсивность равенства
5. Доказать примитивную рекурсивность модуля разности:
6. Доказать примитивную рекурсивность функции «больше»:
7. Доказать примитивную рекурсивность функции «больше или равно»: