Содержание

[Введение 2](#_Toc199256395)

[1. Базовые функции 3](#_Toc199256396)

[2. Примитивная рекурсия 3](#_Toc199256397)

[3. Частично определенная рекурсия 4](#_Toc199256398)

[4. μ-оператор (операция минимизации) 4](#_Toc199256399)

[5. Практическая часть 5](#_Toc199256400)

[5.1. Найти аналитическое выражение функции φ(x) 5](#_Toc199256401)

[5.2. С помощью μ-оператора найти корни уравнения 6](#_Toc199256402)

[5.3. Найти индекс заданного члена ряда Фибоначчи с помощью μ-оператора. 9](#_Toc199256403)

[5.4. Найти число по его факториалу с помощью μ-оператора. 11](#_Toc199256404)

[5.5. Найти минимальное x, удовлетворяющее неравенство 12](#_Toc199256405)

[5.6. Найти минимальное число Фибоначчи, которое больше заданного x 14](#_Toc199256406)

[5.7. Определить μ-оператор для нахождения корней квадратного уравнения 15](#_Toc199256407)

[Заключение 16](#_Toc199256408)

[Список использованной литературы 17](#_Toc199256409)

# Введение

Теория рекурсивных функций является одним из фундаментальных разделов математической логики и теоретической информатики. Она занимается формализацией алгоритмически вычислимых функций и даёт строгую основу для понимания пределов вычислений. В этой области важное место занимает понятие частично определённой рекурсии, которая описывает функции, значение которых может быть не определено на некоторых входах. Это расширяет класс примитивно-рекурсивных функций, где вычисление всегда заканчивается за конечное время.

Центральным элементом теории частично рекурсивных функций выступает μ-оператор (оператор минимизации). Он позволяет описывать вычислительные процессы, в которых осуществляется перебор значений до тех пор, пока не будет найдено первое, удовлетворяющее заданному условию. При этом, если такое значение не существует, функция остаётся неопределённой. Таким образом, μ-оператор формирует переход от всюду определённых функций (примитивных) к частично определённым, расширяя вычислительную мощность формализма, но за счёт потери гарантированной завершённости.

Также важно понимать различие между примитивной рекурсией, где результат вычисления всегда существует и достигается за конечное число шагов, и частичной рекурсией, в которой наличие результата зависит от условий выполнения.

В данной работе будут рассмотрены основные свойства частично рекурсивных функций, а также проведено исследование поведения μ-оператора на конкретных примерах. Цель — проанализировать, как использование минимизации влияет на определённость функций и каким образом она влияет на вычислимость.

# Базовые функции

Базовые функции — это самые простые функции, с которых начинается построение всех частично рекурсивных функций. К ним относятся:

* функция, которая всегда возвращает ноль
* функция, которая увеличивает число на 1 (называется функция наследника)
* функции, которые выбирают один из аргументов (проекционные функции).

Эти функции служат основой для создания более сложных функций с помощью специальных правил и операций.

# Примитивная рекурсия

Примитивная рекурсия — это способ определения функций на натуральных числах, при котором значение функции на аргументе n + 1 выражается через значение функции на n и сам n.

Функция, заданная примитивной рекурсией, строится из базовых функций и двух правил:

1. Значение функции при аргументе 0 задаётся явно (базовый случай).
2. Значение функции при n + 1 вычисляется через значение функции при n с помощью некоторой вычислимой формулы.

Все примитивно рекурсивные функции являются тотальными, то есть определены для всех натуральных чисел и всегда дают результат за конечное число шагов.

Примитивная рекурсия позволяет задавать широкий класс функций, например, сложение, умножение и факториал, при этом гарантируя, что вычисление функции всегда завершится.

# Частично определенная рекурсия

Частично определённая рекурсия — это обобщение примитивной рекурсии, при котором определяемые функции могут быть не полностью определены на всех входных данных. В отличие от примитивно рекурсивных функций, которые всегда дают результат для любого входа, частично рекурсивные функции могут не иметь значения (не определены) для некоторых аргументов.

Это связано с тем, что в построении частично рекурсивных функций используется операция минимизации (μ-оператор), позволяющая находить минимальный аргумент, удовлетворяющий определённому условию. Если такого аргумента нет, вычисление функции не завершается, что приводит к частичной определённости.

Таким образом, частично определённая рекурсия расширяет класс вычислимых функций за пределы примитивной рекурсии и играет важную роль в теории вычислимости и формализации понятия алгоритмической вычислимости.

# μ-оператор (операция минимизации)

μ-оператор — это операция, которая позволяет найти наименьшее натуральное число y, при котором заданная функция f(x, y) принимает значение 0. Формально, если существует такое y, что f(x, y) = 0, то μy.(f(x, y) = 0) — означает минимальное такое y.

Использование μ-оператора расширяет класс вычислимых функций, поскольку с его помощью можно выразить функции, вычисление которых может не завершиться, если подходящего y не существует. Поэтому функции, определённые с помощью μ-оператора, могут быть частично определёнными.

# Практическая часть

## Найти аналитическое выражение функции φ(x)

Задание1. Найти аналитическое выражение функции φ(x), если   
f(x, y) = dif(12, mul(x, y)), и вычислить значение φ(6) с помощью μ-оператора.

Решение:

1. Расшифруем обозначения:

* mul(x, y) – функция умножения x ⋅ y
* dif(a, b) – функция вычитания
* Следовательно, f(x, y) = 12 – x ⋅ y

1. Определим функцию φ(x) через μ-оператор:

По определению, использование μ-оператора означает:

φ(x) = μy.(f(x, y) = 0)

То есть, мы ищем наименьшее y, при котором

12 – x ⋅ y = 0 ⇒ 12 = x ⋅ y

Таким образом:

φ(x) = μy.(x ⋅ y = 12)

1. Найдём аналитическое выражение φ(x):

Нужно найти наименьшее натуральное y, при котором x ⋅ y ≥ 12. Это означает:

y ≥ ⇒ φ(x) = – если делится без остатка, иначе не определена

1. Вычислим значение φ(6):

φ(6) = = 2

Проверим:

f(6, 1) = 12 – 6 ⋅ 1 = 12 – 6 = 6 ≠ 0

f(6, 2) = 12 – 6 ⋅ 2 = 12 – 12 = 0 – условие выполнено

Ответ:

Аналитическое выражение – φ(x) =

Значение – φ(6) = 2

Это задание пример вычислимой, но частично определенной функции: φ(x) определена только для тех x, для которых существует такое y, при котором x ⋅ y = 12.

Например, φ(5) не определена, так как 12 не делится на 5 нацело.

## С помощью μ-оператора найти корни уравнения

Задание 2. С помощью μ-оператор найти корни уравнения:

а) x2 – x – 2 = 0

Решение:

x2 – x – 2 = (x – 2)(x + 1) = 0 ⇒ x = 2, x = –1

Ответ: μy.( x2 – x – 2 = 0) = 2, так как μ-оператор в теории частично рекурсивных функций работает только на множестве натуральных чисел.

б) 2x = x2

Решение:

Преобразуем уравнение к виду:

2x = x2 ⇒2x – x2 = 0

Искомое значение – это наименьшее натуральное x, при котором разность равна нулю. Это можно записать с использованием μ-оператора:

μx.(2x − x2 = 0)

Подставим значения:

x = 0 ⇒ 20 – 02 = 1

x = 1 ⇒ 21 – 12 = 1

x = 2 ⇒ 22 – 22 = 0

Ответ: μx.(2x − x2 = 0)

в) log2x – || = 0

Решение:

Работаем в рамках теории частично рекурсивных функций. Все функции определены на множестве натуральных чисел (включая 0), и используются только допустимые конструкции: элементарные арифметические операции, целочисленные логарифмы, деление без остатка, а также минимизация (μ-оператор).

В уравнении встречаются:

log2x — трактуем как наибольшее целое n, такое что 2n ≤ x

|| — целочисленное деление x на 4 (или же, x div 4)

Перепишем с использование μ-оператора:

μx.(log2x – (x div 4) = 0)

Подставим значения

x = 1 ⇒ log21 – (1 div 4) = 0

Ответ: x = 1

г) x! = (x + 1)2 – 1

Решение:

Для нахождения корней уравнения с помощью μ-оператора необходимо свести его к задаче поиска минимального натурального значения x, при котором выполняется равенство левой и правой частей.

Рассмотрим функцию:

f(x) = |x! – ((x + 1)2 – 1)|

Функция f(x) определена для всех x ∈ N, где x! = 1, если x = 0, или

x! = x ⋅ (x + 1)!, если x > 0.

Абсолютная разность f(x) принимает значение 0 тогда и только тогда, когда x! и (x + 1)2 – 1 равны между собой.

Так как f(x) является частично рекурсивной функцией, мы можем применить μ-оператор:

φ = μx.(f(x) = 0) – наименьшее значение x для решения уравнения.

Вычислим значение f(x) для начальных значений:

|  |  |  |  |
| --- | --- | --- | --- |
| x | x! | (x + 1)2 – 1 | f(x) = |x! – ((x + 1)2 – 1)| |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 3 | 2 |
| 2 | 2 | 8 | 6 |
| 3 | 6 | 15 | 9 |
| 4 | 24 | 24 | 0 |

Таким образом, минимальное значение x, при котором f(x) = 0, равно 4.

Ответ: x = 4.

д) x2 + (x + 1)2 = (x + 2)2

Решение:

Сформулируем функцию, нули которой соответствуют решению уравнения:

f(x) = |x2 + (x + 1)2 – (x + 2)2| = 0

В данном случае все используемые операции (возведение в квадрат, сложение, вычитание) являются примитивно рекурсивными, а значит функция f(x) – частично рекурсивна.

Применим μ-оператор для поиска наименьшего натурального значения x, при котором f(x) = 0:

φ = μx.(|x2 + (x + 1)2 – (x + 2)2| = 0)

Проверим несколько начальных значений:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | x2 | (x – 1)2 | (x + 2)2 | f(x) |
| 0 | 0 | 1 | 4 | |0 + 1 – 4| = 3 |
| 1 | 1 | 4 | 9 | |1 + 4 – 9| = 4 |
| 2 | 4 | 9 | 16 | |4 + 9 – 16| = 3 |
| 3 | 9 | 16 | 25 | |9 + 16 – 25| = 0 |

Ответ: x = 3.

## Найти индекс заданного члена ряда Фибоначчи с помощью μ-оператора.

Задание 3. Найти индекс заданного члена ряда Фибоначчи с помощью  
μ-оператора.

Решение:

Пусть дано некоторое натуральное число a, которое предположительно является элементом ряда Фибоначчи. Требуется найти наименьшее значение  
n ∈ N, такое что:

Fib(n) = a

Ряд Фибоначчи можно задать рекурсивно:

Fib(0) = 0; Fib(1) = 1; Fib(n + 2) = Fib(n + 1) + Fib(n)

Эта функция является примитивно-рекурсивной, так как построена с помощью базовых операций и рекурсии над натуральными числами.

Рассмотрим частично определённую функцию:

f(n, a) = |Fib(n) – a|

Она возвращает 0, если Fib(n) = a, и положительное значение в противном случае.

Используем μ-оператор для поиска наименьшего n, при котором  
f(n, a) = 0:

φ(a) = μn.(|Fib(n) – a| = 0)

φ(a) — частично определённая функция, возвращающая индекс n, если такой существует. Если a не принадлежит ряду Фибоначчи, то μ-оператор не завершится, и функция останется неопределённой для этого аргумента.

Рассмотрим пример, найдем индекс числа 8:

|  |  |
| --- | --- |
| n | Fib(n) |
| 0 | 0 |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 5 |
| 6 | 8 |
| 7 | 13 |
| 8 | 21 |

Fib(8) = 21 ⇒ φ(21) = 8

Ответ: φ(a) = μn.(|Fib(n) – a| = 0)

## Найти число по его факториалу с помощью μ-оператора.

Задание 4. Найти число по его факториалу с помощью μ-оператора.

Решение:

Пусть задано натуральное число a, которое предполагается равным n! для некоторого n ∈ N. Требуется найти наименьшее n, такое что:

n! = a

Функция факториала задается рекурсивно:

fact(0) = 1;

fact(n + 1) = (n + 1) ⋅ fact(n)

Это функция – примитивно-рекурсивная, так как строится с помощью композиции и рекурсии.

Построим частично определенную функцию:

f(n, a) = |fact(n) – a|

Она равна 0 тогда и только тогда, когда fact(n) = a.

Теперь с помощью μ-оператор ищем наименьшее n, при котором функция обращается в 0:

φ(a) = μn.(|fact(n) – a| = 0)

Эта функция частично определена: она определена только для таких a, которые действительно являются факториалами некоторых n ∈ N.

Рассмотрим пример, найдем n, если a = 120:

|  |  |
| --- | --- |
| n | fact(n) |
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 6 |
| 4 | 24 |
| 5 | 120 |

φ(120) = 5

Ответ: φ(a) = μn.(|fact(n) – a| = 0)

Если a не является факториалом никакого натурального числа (например, a = 7), то функция φ(a) остается неопределенной, поскольку цикл поиска по n никогда не завершится. Это подчеркивает частичную определенность данной функции.

## Найти минимальное x, удовлетворяющее неравенство

Задание 5. Найти минимальное x, удовлетворяющее неравенство x! > 2x с помощью μ-оператора.

Решение:

Это задача на поиск минимального значения, при котором выполняется некоторое неравенство. Мы можем воспользоваться μ-оператором, применив его к частично определённой функции.

Определим функцию f(x):

f(x) =

Чтобы задать это в рамках μ-оператора, удобно определить

f(x) = |2x – x!| – (2x – x!)

Объяснение:

* Если x! > 2x, то 2x – x! < 0, и

|2x – x!| – (2x – x!) = –(2x – x!) – (2x – x!) = –2(2x – x!) > 0

* Если x! ≤ 2x, то выражение будет равно 0 или положительному числу, но не будет соответствовать условию задачи x! > 2x.

Для корректной подстановки под μ-оператор нам нужно:

g(x) =

Тогда: φ(x) = μx.(g(x) = 0)

Тогда окончательная функция с использованием μ-оператора будет иметь вид:

φ() = μx.(x! > 2x)

Формальная запись будет иметь следующий вид:

φ() = μx.(x! > 2x)

Так как неравенство невозможно выразить напрямую, мы можем воспользоваться эквивалентной функцией:

φ() = μx.(| 2x – x!| – (2x – x!) > 0)

Посчитаем значения:

|  |  |  |
| --- | --- | --- |
| x | x! | 2x |
| 0 | 1 | 1 |
| 1 | 1 | 2 |
| 2 | 2 | 4 |
| 3 | 6 | 8 |
| 4 | 24 | 16 |

При x = 4, x! = 24, 24 = 16, 24 > 16.

Ответ: x = 4

## Найти минимальное число Фибоначчи, которое больше заданного x

Задание 6. Найти минимальное число Фибоначчи, которое больше заданного x, с помощью μ-оператора.

Решение:

Пусть F(n) — функция, возвращающая n-е число Фибоначчи, определённая рекурсивно:

F(0) = 0, F(1) = 1, F(n) = F(n − 1) +F(n − 2) для n ≥ 2.

Для того, чтобы представить функцию с использованием μ-оператора, представим задачу в виде функции, для которой нужно найти минимальное n, при котором F(n) > x.

f(x, n) =

Тогда искомая функция будет выглядеть как:

φ() = F(μn.(f(x, n) = 0)), где μn.(f(x, n) = 0) — минимальное n,  
для которого F(n) > x

Функция φ(x) возвращает **минимальное число Фибоначчи**, которое строго больше заданного числа x.

Пример: найти минимальное число Фибоначчи больше x = 10.

Рассмотрим следующие числа Фибоначчи:

F(0) = 0

F(1) = 1

F(2) = 1

F(3) = 2

F(4) = 3

F(5) = 5

F(6) = 8

F(7) = 13

Мы ищем минимальный n, что F(n) > 10:

* F(6) = 8 ≤ 10 — не подходит
* F(7) = 13 > 10 — подходит

Ответ: φ() = F(μn.(f(x, n) = 0))

## Определить μ-оператор для нахождения корней квадратного уравнения

Задание 7. Определить μ-оператор для нахождения корней квадратного уравнения ax2 + bx + c = 0, где a, b, c > 0. Такой вид уравнения является условием возможного наличия натуральных корней, исходя из равенства  
(x – x1)(x – x2) = x2 – (x1 + x2)x + x1x2 = 0 для x1, x2 > 0. Для тестирования  
μ-оператора использовать наборы коэффициентов при указанных корнях:

а) уравнение: (x – 1)(x – 1) = x2 – 2x + 1 = 0. Коэффициенты: a = 1, b = 2,  
c = 1

# Заключение

В ходе выполнения контрольной работы была рассмотрена тема частично определённой рекурсии, которая лежит в основе теории частично рекурсивных функций — важного раздела математической логики и теории вычислимости. Основное внимание было уделено изучению μ-оператора, как ключевого инструмента, позволяющего расширить класс вычислимых функций за пределы примитивной рекурсии. Благодаря минимизации стало возможным определение функций, значение которых не всегда существует для всех входных данных, что приводит к частичной определённости.

В работе были решены практические задачи с использованием μ-оператора, включая нахождение корней уравнений, индексов и значений в известных числовых последовательностях (факториал, числа Фибоначчи), а также задачи на неравенства. Это позволило закрепить понимание принципов построения частично рекурсивных функций и показать, как теоретические конструкции применяются на практике.

Таким образом, проведённое исследование демонстрирует как ограничения примитивной рекурсии, так и силу μ-оператора в контексте расширения понятия вычислимости. Частично определённая рекурсия служит важным шагом к более общему пониманию алгоритмов и их пределов.

# Список использованной литературы

1. Игошин В.И. “Математическая логика: учебное пособие”. – М.: ИНФРА-М, 2019. – 398 с.
2. Пруцков А.В., Волкова Л.Л. “Математическая логика и теория алгоритмов: учебник”. – М.: ИНФРА-М, 2018. – 152 с.
3. Колмогоров А.Н., Драгалин А.Г. “Математическая логика. Изд. 3-е, стереотипное”. – М.: КомКнига, 2006, 240 с.
4. Игошин В.И. “Сборник задач по математической логике и теория алгоритмов: учебное пособие”. – М.: ИНФРА-М, 2019. – 392 с.