

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Программная инженерия
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №7
Исчисления и абстрактная интерпретация

тема

Преподаватель

А. С. Кузнецов

инициалы, фамилия

подпись, дата

Студент КИ23-16/16, 032318988
номер группы, зачётной книжки

Е. А. Александров

инициалы, фамилия

подпись, дата

Красноярск 2025

1.1 Цель

Исследование проблем вычислимости без использования абстрактной машины Тьюринга.

1.2 Задачи

1. Ознакомиться с теоретическими сведениями по проблемам вычислимости и разрешимости, а также метода абстрактной интерпретации.

2. Получить у преподавателя собственный вариант задания, предусматривающего построение вычислителя заданной функции над целыми числами (1 часть), а также проведение абстрактной интерпретации (2 часть).

3. Используя изученные механизмы, произвести программную реализацию вычислителя заданной математической функции для заданных аргументов, причем исключительно средствами примитивной и частичной рекурсии. Области определения и значений не обязательно совпадают с множеством натуральных чисел, могут быть ограничены "компьютерными числами" (например, Integer.MAX_VALUE и Integer.MIN_VALUE). Это следует учитывать при конструировании вычислителя.

4. Используя метод абстрактной интерпретации, для произвольной (на выбор и вкус студента) подпрограммы определить знаки всех переменных. Минимальное количество строк кода процедуры без учета комментариев не менее 10.

5. Написать отчет и представить его к защите вместе с исходным кодом программы. Защита может проводиться как в аудитории, так и в исключительных случаях дистанционно.

1.3 Задание

В части 1 необходимо произвести программную реализацию вычислителя заданной математической функции для заданных аргументов, причем исключительно средствами примитивной и частичной рекурсии, или формально доказать невозможность этого. Привести примеры выполнения вычислений.

В части 2 необходимо, используя метод абстрактной интерпретации, для произвольной программной процедуры с количеством строк кода без комментариев не менее 10, определить знаки всех переменных.

Мною был получен вариант 2:

$$f(x, y) = x^y, \text{ где } ^\wedge \text{ — это операция возведения в степень, а } y \geq 0.$$

2 ХОД РАБОТЫ

2.1 Задание 1

Программная реализация вычислителя функции для моего варианта представлена на рисунках 1-3.

```
class BasePrimitiveFunctions:
    """
    Три базовые примитивно рекурсивные функции:
    1. Нулевая функция Z (или 0)
    2. Функция следования S
    3. Функция проекции I^n_k
    """

    @staticmethod
    def zero(*args):
        """
        Нулевая функция: Z() = 0 или 0^n(x_1, ..., x_n) = 0
        Возвращает 0 для любых аргументов
        """
        return 0

    @staticmethod
    def succ(n):
        """
        Функция следования: S(n) = n + 1
        """
        return n + 1

    @staticmethod
    def proj(k, *args):
        """
        Функция проекции: I^n_k(x_1, ..., x_n) = x_k
        Возвращает k-й аргумент

        Пример: proj(1, a, b, c) = b
        """
        return args[k]

class PrimitiveRecursion(BasePrimitiveFunctions):
    """
    Функции, построенные через ПРИМИТИВНУЮ РЕКУРСИЮ
    """

    def pred(self, n):
        """
        Предшественник через ПРИМИТИВНУЮ РЕКУРСИЮ:
        pred(0) = 0 = Z()
        pred(S(n)) = n = I^2_0(n, pred(n))
        """
        if n == 0:
            return self.zero()
```

Рисунок 1 - Программная реализация вычислителя функции, часть 1

```

46     else:
47         prev_result = self.pred(n - 1)
48         return self.proj(0, n - 1, prev_result)
49
50     def add(self, x, y):
51         """
52             Сложение через ПРИМИТИВНУЮ РЕКУРСИЮ:
53             add(x, 0) = x = I^1_0(x)                  -- базовый случай
54             add(x, S(y)) = S(add(x, y))            -- рекурсивный шаг
55         """
56         if y == 0:
57             return self.proj(0, x)
58         else:
59             return self.succ(self.add(x, y - 1))
60
61     def mult(self, x, y):
62         """
63             Умножение через ПРИМИТИВНУЮ РЕКУРСИЮ:
64             mult(x, 0) = 0 = Z()                    -- базовый случай
65             mult(x, S(y)) = add(mult(x, y), x)    -- рекурсивный шаг
66         """
67         if y == 0:
68             return self.zero()
69         else:
70             return self.add(self.mult(x, y - 1), x)
71
72     def power(self, x, y):
73         """
74             Возведение в степень через ПРИМИТИВНУЮ РЕКУРСИЮ:
75             power(x, 0) = 1 = S(Z())                -- базовый случай
76             power(x, S(y)) = mult(power(x, y), x)  -- рекурсивный шаг
77         """
78         if y == 0:
79             return self.succ(self.zero()) # S(Z()) = 1
80         else:
81             return self.mult(self.power(x, y - 1), x)
82
83
84     class Calculator(PrimitiveRecursion):
85         """Функции, построенные через СУПЕРПОЗИЦИЮ (композицию)"""
86
87     def square(self, x):
88         """

```

Рисунок 2 - Программная реализация вычислителя функций, часть 2

```

def square(self, x):
    """
    g(x) = x2 через СУПЕРПОЗИЦИЮ
    """
    return self.mult(self.proj(0, x), self.proj(0, x))

def power_self(self, x):
    """
    f(x) = xx через СУПЕРПОЗИЦИЮ
    """
    return self.power(self.proj(0, x), self.proj(0, x))

def cube(self, x):
    """
    c(x) = x3 через СУПЕРПОЗИЦИЮ
    """
    return self.mult(self.square(x), self.proj(0, x))

if __name__ == "__main__":
    calc = Calculator()
    print("Вычислитель xy")
    examples = [(4, 3), (2, 10), (3, 4)]
    for x, y in examples:
        result = calc.power(x, y)
        print(f" {x}>4} ^ {y}<4} = {result}>10}")

    print("Вычислитель xx")
    for x in (1, 2, 3, 4):
        result = calc.power_self(x)
        print(f" {x}>4} ^ {x}<4} = {result}>10}")

```

Рисунок 3 - Программная реализация вычислителя функции, часть 3

Пример работы вычислителя представлен на рисунке 4.

```

PS D:\projects\university\theory_of
    4 ^ 3      =      64
    2 ^ 10     =     1024
    3 ^ 4      =      81

```

Рисунок 4 - Пример работы вычислителя

2.2 Часть 2

Необходимо, используя метод абстрактной интерпретации, для произвольной программной процедуры с количеством строк кода без комментариев не менее 10, определить знаки всех переменных.

Для анализа выбрана функция, представленная на рисунке 5.

```
def some_function(a, b):
    c = a + b
    d = a * c
    e = b - a
    f = c * d
    g = a + d
    h = e * f
    i = g - b
    j = h + i
    k = d * e
    result = j + k
    return result
```

Рисунок 5 - Исходный код анализируемой функции

Условные обозначения, использованные при анализе:

+: переменная принимает значения строго больше нуля

-: переменная принимает значения строго меньше нуля

0: переменная принимает только значение 0

Unknown: переменная принимает любые значения {-;0;+}

Анализ знаков переменных при a: +, b: +

c: +

d: +

e: Unknown

f: +

g: +

h: Unknown

i: Unknown

j: Unknown

k: Unknown

result: Unknown

Анализ знаков переменных при a: +, b: -

c: Unknown

d: Unknown

e: -

f: Unknown

g: Unknown

h: Unknown

i: Unknown

j: Unknown

k: Unknown

result: Unknown

Анализ знаков переменных при a: -, b: +

c: Unknown

d: Unknown

e: +

f: Unknown

g: Unknown

h: Unknown

i: Unknown

j: Unknown

k: Unknown

result: Unknown

Анализ знаков переменных при a: -, b: -

c: -

d: +

e: Unknown

f: -

g: Unknown

h: Unknown

i: Unknown
j: Unknown
k: Unknown
result: Unknown

Анализ знаков переменных при a: 0, b: +

c: +
d: 0
e: +
f: 0
g: 0
h: 0
i: -
j: -
k: 0

result: -

Анализ знаков переменных при a: +, b: 0

c: +
d: +
e: -
f: +
g: +
h: -
i: +
j: Unknown
k: -

result: Unknown

Анализ знаков переменных при a: +, b: 0

c: +
d: +
e: -
9

f: +

g: +

h: -

i: +

j: Unknown

k: -

result: Unknown

Анализ знаков переменных при a: 0, b: -

c: -

d: 0

e: -

f: 0

g: 0

h: 0

i: +

j: +

k: 0

result: +

Анализ знаков переменных при a: -, b: 0

c: -

d: +

e: +

f: -

g: Unknown

h: -

i: Unknown

j: Unknown

k: +

result: Unknown

Анализ знаков переменных при a: 0, b: 0

c: 0

d: 0

e: 0

f: 0

g: 0

h: 0

i: 0

j: 0

k: 0

result: 0

Вывод

По результатам работы был изучен теоретический материал по теме «Исчисления и абстрактная интерпретация». Все поставленные цели и задачи были выполнены. Задания были выполнены и помогли лучше усвоить пройденный материал.