

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №7

Исчисления и абстрактная интерпретация

тема

Преподаватель

подпись, дата

А. С. Кузнецов
инициалы, фамилия

Студент КИ18-17/16 031830504

номер группы, зачетной книжки

подпись, дата

Е.В. Железкин
инициалы, фамилия

Красноярск 2021

1 Цель работы

Исследование проблем вычислимости без использования абстрактной машины Тьюринга.

2 Задача работы

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 определить знаки всех переменных.

Вариант (15)

$$f(x) = 3^{x^3}$$

2.1 Инструкция по запуску

Необходимо установить *python*, желательно версии 3 и выше (выполнено на версии 3.9.4):

- Страница загрузки для Windows: <https://www.python.org/downloads/>
- Для Linux есть несколько способов, один из них инструмент apt-get:

```
$ sudo apt-get update
```

```
$ sudo apt-get install python3.8
```
- Или загрузить, распаковать и установить образ:

```
$ wget https://www.python.org/ftp/python/3.8.2/Python-3.8.2.tgz
```

```
$ tar -xvf Python-3.8.2.tgz
```

Для следующего шага понадобится компилятор gcc, но, думаю, это не проблема. Переходим в распакованную папку и собираем+устанавливаем:

```
$ cd Python-3.8.2
$ ./configure
$ make
$ sudo make install
```

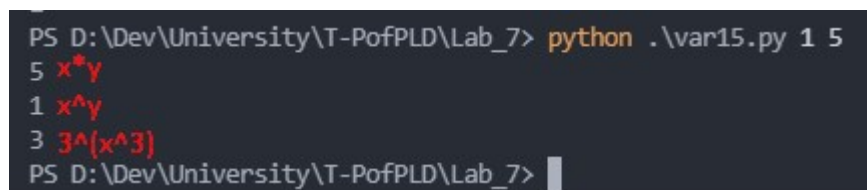
Далее на любой из двух систем выполнить:

```
$ python var15.py <argv>
```

3 Ход работы

Часть 1

Произведена программная реализация вычислителя математической функции $f(x) = 3^{x^3}$:



```
PS D:\Dev\University\T-PofPLD\Lab_7> python .\var15.py 1 5
5 x*y
1 x^y
3 3^(x^3)
PS D:\Dev\University\T-PofPLD\Lab_7>
```

Рисунок 1 – Результаты работы программы с параметрами «1 5»

При вызове моей функции для аргумента $|x| > 1$ происходит переполнение рекурсивного стека(Рисунок 2). При увеличении глубины функция обрабатывает некорректно - ничего не выводится (Рисунок 3):

```

PS D:\Dev\University\T-PofPLD\Lab_7> python .\var15.py 2 5
Traceback (most recent call last):
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 50, in <module>
    main(int(sys.argv[1]), int(sys.argv[2]))
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 46, in main
    print(var15_with_minus(f_arg))
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 39, in var15_with_minus
    )) if x < 0 else pow_with_prf(3, pow_with_prf(x, 3))
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 26, in pow_with_prf
    def pow_with_prf(x, y): return prf(
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 19, in <lambda>
    def prf(f): return lambda x, y, z: z if y == 0 else prf(
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 19, in <lambda>
    def prf(f): return lambda x, y, z: z if y == 0 else prf(
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 19, in <lambda>
    def prf(f): return lambda x, y, z: z if y == 0 else prf(
[Previous line repeated 3 more times]
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 20, in <lambda>
    f)(x, pred(y), f(x, z))
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 27, in <lambda>
    lambda xt, yt: mult_with_prf(xt, yt))(x, y, 1)
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 23, in mult_with_prf
    def mult_with_prf(x, y): return prf(lambda xt, yt: add(xt, yt))(x, y, 0)
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 19, in <lambda>
    def prf(f): return lambda x, y, z: z if y == 0 else prf(
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 19, in <lambda>
    def prf(f): return lambda x, y, z: z if y == 0 else prf(
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 19, in <lambda>
    def prf(f): return lambda x, y, z: z if y == 0 else prf(
[Previous line repeated 243 more times]
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 20, in <lambda>
    f)(x, pred(y), f(x, z))
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 23, in <lambda>
    def mult_with_prf(x, y): return prf(lambda xt, yt: add(xt, yt))(x, y, 0)
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 13, in add
    def add(x, y): return x if y == 0 else add(x, y - 1) + 1
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 13, in add
    def add(x, y): return x if y == 0 else add(x, y - 1) + 1
  File "D:\Dev\University\T-PofPLD\Lab_7\var15.py", line 13, in add
    def add(x, y): return x if y == 0 else add(x, y - 1) + 1
[Previous line repeated 736 more times]
RecursionError: maximum recursion depth exceeded in comparison
PS D:\Dev\University\T-PofPLD\Lab_7> █

```

Рисунок 2 – Результаты работы программы с параметрами «2 5»

```
def main(f_arg, s_arg):
    print(mult_with_prf(f_arg, s_arg))
    print(pow_with_prf(f_arg, s_arg))
    sys.setrecursionlimit(3000)
    print(var15_with_minus(f_arg))

if __name__ == '__main__':
    main(int(sys.argv[1]), int(sys.argv[2]))
```

MS OUTPUT TERMINAL DEBUG CONSOLE

PowerShell 10

PowerShell 32

PS D:\Dev\University\T-PofPLD\Lab_7> python .\var15.py 2 5

10

32

PS D:\Dev\University\T-PofPLD\Lab_7> █

Рисунок 3 – Результаты работы программы с параметрами «2 5»

```
def main(f_arg, s_arg):
    # print(mult_with_prf(f_arg, s_arg))
    # print(pow_with_prf(f_arg, s_arg))
    sys.setrecursionlimit(3000)
    print(var15_with_minus(f_arg))
```

MS OUTPUT TERMINAL DEBUG CONSOLE

PowerShell 10

PowerShell 32

PS D:\Dev\University\T-PofPLD\Lab_7> python .\var15.py 2 5

PS D:\Dev\University\T-PofPLD\Lab_7> python .\var15.py -1 5

1/3

Рисунок 4 – Результаты работы программы с параметрами «-1 5»

Часть 2

Имеем домен $S = \{minus, zero, plus, unknown, NaN\}$, где *minus* – отрицательные числа; *zero* – нуль, *plus* – положительные, *unknown* – неизвестно, *infinity* – числовой эквивалент бесконечности.

Если $A \in S$ и $B \in S$, то

$R = A / B$:

- *zero*, если $A = zero$ и $B = zero$;
- *plus*, если $A = plus$ и $B = plus$ или $A = minus$ и $B = minus$;
- *minus*, если $A = minus$ и $B = plus$ или $A = plus$ и $B = minus$;
- *unknown*, если $A = unknown$ и $B = unknown$;
- *infinity*, если $B = zero$;

Произвольная программная процедура:

$a = 3$ (*plus*)

$b = -1$ (*minus*)

$c = a / b$ (*minus*)

$d = 0 / 1$ (*zero*)

$e = 0 / 0$ (*infinity*)

$f = 0 + z$ (*unknown*)

4 Вывод

В ходе данной лабораторной работы были исследованы проблемы вычислимости без использования абстрактной машины Тьюринга.

ПРИЛОЖЕНИЕ А

Листинг 1 – файл var15.py

```
import sys

def z(x): return 0

def pred(x): return 0 if x == 0 else x - 1

def subtr(x, y): return x if y == 0 else pred(subtr(x, y - 1))

def add(x, y): return x if y == 0 else add(x, y - 1) + 1

def mult(x, y): return add(x, mult(x, y - 1))

def prf(f): return lambda x, y, z: z if y == 0 else prf(
    f)(x, pred(y), f(x, z))

def mult_with_prf(x, y): return prf(lambda xt, yt: add(xt, yt))(x, y, 0)

def pow_with_prf(x, y): return prf(
    lambda xt, yt: mult_with_prf(xt, yt))(x, y, 1)

# RecursionError: maximum recursion depth exceeded in comparison if |x| > 1
# but it should work right
# for all power functions it is indicated that the exponent is greater than zero, I
# will assume that for mine too
# prf - primitive recursive function
def var15(x): return pow_with_prf(3, pow_with_prf(x, 3))

# crutch
def var15_with_minus(x): return '1/' + str(pow_with_prf(3, pow_with_prf(-x, 3)
    )) if x < 0 else pow_with_p
rf(3, pow_with_prf(x, 3))

def main(f_arg, s_arg):
```

```
# print(mult_with_prf(f_arg, s_arg))
# print(pow_with_prf(f_arg, s_arg))
print(var15_with_minus(f_arg))

if __name__ == '__main__':
    main(int(sys.argv[1]), int(sys.argv[2]))
```