Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 7 по дисциплине «Функциональные и логические языки программирования»

Тема Лисп. Рекурсивные выражения.

Студент Одинцов Е.В.

Группа ИУ7-53БВ

Преподаватели Строганов Ю.В.

Москва, 2024

Содержание

ВВЕДЕНИЕ

Целью данной лабораторной работы является исследование особенностей использования рекурсии для вычисления чисел Фибоначчи и факториала на языке программирования Common Lisp. В работе рассматриваются два типа рекурсии: хвостовая и обычная, и приводятся примеры функций для каждого типа.

1 Аналитическая часть

Функции Фибоначчи и факториала имеют широкое применение в различных областях, таких как математика, программирование и алгоритмы. Рекурсивные алгоритмы для их вычисления позволяют наглядно продемонстрировать применение рекурсии и её особенности.

1.0.1 Функция Фибоначчи

Числа Фибоначчи $F(n)$ определяются рекурсивной формулой:

$$F(n) = F(n-1) + F(n-2), \tag{1.1}$$

где $F(0) = 0$ и $F(1) = 1$.

1.0.2 Функция факториала

Факториал числа n, обозначаемый как $n!$, определяется произведением всех целых чисел от 1 до n:

$$n! = n \times (n-1) \times \cdots \times 1, \tag{1.2}$$

где $0! = 1$.

Обычная рекурсия для этих функций проста, но менее оптимальна, так как при больших значениях n может привести к переполнению стека вызовов. Хвостовая рекурсия более эффективна за счёт того, что сохраняет результат в аккумуляторе, что позволяет компилятору оптимизировать выполнение.

2 Технологическая часть

В этом разделе представлены реализации функций для вычисления чисел Фибоначчи и факториала на языке Common Lisp с использованием как хвостовой, так и обычной рекурсии.

2.0.1 Реализация функции Фибоначчи

Обычная рекурсия:

Листинг 2.1 — Функция Фибоначчи с обычной рекурсией

```
(defun fibonacci (n)
  (if (<= n 1)
      n
      (+ (fibonacci (- n 1)) (fibonacci (- n 2)))))
```

Хвостовая рекурсия:

Листинг 2.2 — Функция Фибоначчи с хвостовой рекурсией

```
(defun fibonacci-tail (n &optional (a 0) (b 1))
  (if (= n 0)
      a
      (fibonacci-tail (- n 1) b (+ a b))))
```

В первой функции ‘fibonacci‘ результат вычисляется после всех рекурсивных вызовов, что делает её обычной рекурсией. Во второй функции ‘fibonacci-tail‘ результат вычисляется в аккумуляторе и передаётся на каждом шаге, что делает её хвостовой.

2.0.2 Реализация функции факториала

Обычная рекурсия:

Листинг 2.3 — Функция факториала с обычной рекурсией

```
(defun factorial (n)
  (if (<= n 1)
      1
      (* n (factorial (- n 1)))))
```

Хвостовая рекурсия:

Листинг 2.4 — Функция факториала с хвостовой рекурсией

```
(defun factorial-tail (n &optional (acc 1))
```

```
    (if (<= n 1)
        acc
        (factorial-tail (- n 1) (* acc n))))
```

В функции ‘factorial‘ используется обычная рекурсия, где вычисление происходит после возвращения значений из всех вложенных вызовов. В функции ‘factorial-tail‘ результат передаётся через аккумулятор ‘acc‘, что делает её хвостовой.

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе были исследованы два подхода к рекурсивному вычислению чисел Фибоначчи и факториала: обычная и хвостовая рекурсия. Хвостовая рекурсия более оптимальна, так как позволяет использовать оптимизацию компилятора и избегать переполнения стека вызовов. Однако её использование требует добавления аккумулятора для накопления промежуточных результатов.

Таким образом, хвостовая рекурсия предпочтительна для вычислений, требующих большого количества рекурсивных вызовов, так как позволяет снизить затраты ресурсов и повысить производительность программы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Graham, P. (1995). *ANSI Common Lisp*. Prentice Hall.

2. Wikipedia. Tail call. `https://en.wikipedia.org/wiki/Tail_call`. [Дата обращения: октябрь 2024].