Отчет по лабораторной работе № 2 по курсу «Функциональное программирование»

Студент группы М8О-307 МАИ *Безлуцкая Елизавета*, №2 по списку
Контакты: lizabezlutskaya@gmail.com
Работа выполнена: 28.03.2019

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

Тема работы

Простейшие функции работы со списками Коммон Лисп.

Цель работы

Научиться конструировать списки, находить элемент в списке, использовать схему линейной и древовидной рекурсии для обхода и реконструкции плоских списков и деревьев.

Задание (вариант №11)

Дан список действительных чисел $(x_1 \ ... \ x_n)$, $n \geqslant 2$. Запрограммируйте рекурсивно на языке Коммон Лисп функцию, вычисляющую выражение вида:

$(x_1 + x_n) * (x_2 + x_{n-1}) * ... * (x_n + x_1)$.

Оборудование студента

Процессор Intel Core i5-3230M 4 @ 2.6GHz, память: 350Gb, разрядность системы: 64.

Программное обеспечение

Ubuntu 16.04 LTS, clisp compiler

Идея, метод, алгоритм

Функция `product-sum2` вызывает вспомогательную функцию `sub-product-sum2` со следующими аргументами: списком `l` и перевёрнутым списком `l`. `sub-product-sum2` рекурсивна и работает следующим образом:

- если список `l` пуст, функция вернет 1, иначе:
- перемножаем сумму первых элементов списков и результат рекурсивного вызова функции `sub-product-sum2` с аргументами в виде хвостов текущих списков.

Сценарий выполнения работы

Распечатка программы и её результаты

Исходный код

```
(defun sub-product-sum (l1 l2)
    (cond ((null l1) 1)
          (t (* (+ (first l1)(first l2))(sub-product-sum (rest
   l1)(rest l2))))))

(defun product-sum2 (l)
    (sub-product-sum l (reverse l)))
```

Результаты работы

```
(print (product-sum2 '(1 2 3 4 5)))
7776
(print (product-sum2 '(0 0)))
0
(print (product-sum2 '(1 -1)))
0
(print (product-sum2 '(1 0 1 1)))
4
(print (product-sum2 '(-8 5)))
9
(print (product-sum2 '(-8 -5)))
169
(print (product-sum2 '(-8 -5 5)))
-90
(print (product-sum2 '(1 3 -5 4 6 8 -2 4 9 4 0)))
12845056
```

Замечания автора по существу работы

С хвостовой рекурсией я познакомилась, когда проходила курс «Логическое программирование» и писала код лабораторных работ на языке Prolog. В связи с этим трудностей у меня не возникло.

Выводы

Первоначально задуманное решение не требовало дополнительной памяти, но требовало отсечения последнего элемента списка. Варианты реализации такой функции мне показались не очень привлекательными, поэтому я изменила тактику. В итоге программа работает за линейное время и память.