

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

Студент группы 8О-308 МАИ *Балес Александр*, №3 по списку
Контакты: `aleks_bales@mail.ru`
Работа выполнена: 01.04.2016

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

1. Тема работы

Простейшие функции работы со списками Коммон Лисп.

2. Цель работы

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

3. Задание(вариант 2.37)

Запрограммируйте на языке Коммон Лисп функционал `map-set (f X)`, аргументами которого являются функция одного аргумента `f` и список `X`, рассматриваемый как множество. Результатом вызова должно быть множество из результатов применения `f` к каждому из элементов `X`. В списки, представляющие множества, нет повторений, а порядок элементов не имеет значения

4. Оборудование студента

Процессор Intel Core i5-3210 4@2.5GHz, память: 8192Mb, разрядность системы: 64.

5. Программное обеспечение

ОС Ubuntu 14.04, среда GNU Common Lisp 2.6.10

6. Идея, метод, алгоритм

Написал две вспомогательных ф-ии, одна из которых -- `iterator`, реализация итерации по списку при помощи рекурсии ("откусывать" голову списка, и рекурсивно вызывать

для хвоста); а вторая ф-ия -- `comp-elem`, пробегается каждый раз с текущего места в списке и до конца, проверяя на совпадение элементы, причемы перед проверкой применяется ф-ия `f`, которая передается в качестве формального параметра.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

```
;; var 2.37 (level 3)
;; program was developed by Alexander Bales 80-308

(defun iterator (arrList)
  (if (null arrList) ()
      (let ((cur (first arrList)))
        (if (not (comp-elem cur (rest arrList)))
            (cons cur (iterator (rest arrList)))
            (iterator (rest arrList))))))

(defun comp-elem (elem cur-list)
  (cond ((null cur-list) NIL)
        ((equal elem (first cur-list)) T)
        ((comp-elem elem (rest cur-list)))))

(defun genList (f arr)
  (if (null arr) ()
      (cons (funcall f (first arr)) (genList f (rest arr)))))

(defun map-set (f x)
  (iterator (genList f x)))

(print (map-set #'identity (list 20 20 30 30 40 40)))
(print (map-set #'abs '(1 2 -3 -2)))
```

8.1. Результаты

```
alex31@ProBook-4540s:~/src/GitHub/FunctionalProgramming/labs/lab2$ gcl -load lr2.l  
(20 30 40)  
(1 3 2) GCL (GNU Common Lisp) 2.6.10 CLtL1 Apr 2 2014 14:22:53  
Source License: LGPL(gcl,gmp), GPL(unexec,bfd,xgcl)  
Binary License: GPL due to GPL'ed components: (XGCL READLINE UNEXEC)  
Modifications of this banner must retain notice of a compatible license  
Dedicated to the memory of W. Schelter  
  
Use (help) to get some basic information on how to use GCL.  
Temporary directory for compiler files set to /tmp/  
  
alex31@ProBook-4540s:~/src/GitHub/FunctionalProgramming/labs/lab2$
```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
07.04.2016	Оптимизация	Добавил ф-ию genList , которая возвращает список из исходных элементов, применяя к ним ф-ию, передаваемую в map-set первым аргументом.	Асимптотически сложность не изменилась, но по времени должно работать быстрее

10. Замечания, выводы

Из распечатки программы сразу видно, что сложность программы по времени $O(N^2)$, где N -- число элементов в списке. Стоит сразу также отметить, что ф-ии **iterator**, **comp-elem** реализованы при помощи концевой рекурсии, а это значит, что скорее всего, компилятор Коммон Лисп их оптимизирует до простого цикла. По памяти так же сложность составляет $O(N^2)$, т.к. мы при каждом рекурсивном вызове передаем в качестве параметром списки длины $N+1$ генерируется новый список из неповторяющихся элементов.