



Министерство образования Российской Федерации
Московский Государственный Технический Университет
им. Н.Э. Баумана

Отчет по лабораторной работе №3
По курсу «Функциональное и логическое
программирование»

Студент
Группа
Преподаватель

Медведев А.В.
ИУ7-62
Толпинская Н.Б.

Теоретическая часть

Базис Lisp.

- Базис - атомы и структуры (бинарные узлы), хранящие два указателя.
- Базис включает базисные функции и базисные функционалы (atom, eq, car, cdr, cons, quote, cond, lambda, label, eval)
- Над базисом строятся простые формулы в виде списков, где первый аргумент является именем функции, а остальные аргументы ее аргументы.
- Базис это совокупность языка, без которого нет языка (минимальный набор средств языка)

Классификация функций в Lisp.

1. Чистые или строго-математические (Имеют фиксированное кол-во аргументов и один результат)
2. Специальные (формы) (Могут иметь произвольное количество аргументов; Не все аргументы обрабатываются одинаково)
3. Псевдо-функции (Функции обеспечивающие некоторый эффект аппаратуры)
4. Функции допускающие вариационные значения (Позволяют реализовывать логические преобразования)
5. Функции, допускающие ленивые вычисления (Вызов функций сводится к представлению правила выполнения в зависимости от контекста)
6. Функции высших порядков (функционалы) (Используются для синтаксического конструирования программ; В качестве аргументов используются функциональные объекты)

Как представляются списки в ОП?

Список - это особый вид S-выражения, который может быть пустым/не пустым, если он не пустой, то он имеет первый элемент (голову) и хвост (является списком). S-выражения представлены в виде точечных пар, которые состоят из унифицированных структур - блоков памяти - бинарных узлов. Каждый бинарный узел имеет небольшой объем, достаточный для хранения двух типизированных указателей (CAR и CDR, левый и правый, голова и хвост). Пара из первого элемента списка ("голова") и остальных элементов списка ("хвост") представляют собой пару указателей на атомы

Как выполняются функции CAR и CDR, какие результаты они вернут в разных случаях?

Базовыми функциями доступа к данным в lisp являются функции CAR и CDR.

- CAR - В качестве аргумента принимает список (1 аргумент) и возвращает первый элемент списка, если список не пустой и Nil - иначе
- CDR - в качестве аргумента принимает список (1 аргумент) и возвращает список, состоящий из всех элементов кроме первого, если список не пустой и Nil - иначе

Отличие работы list и cons

- CONS - в качестве аргумента принимает два аргумента, каждый из которых S-выражение. Возвращает: 1) точечную пару, если второй аргумент не список; 2) Новый список, если второй аргумент список, а первый аргумент добавляет в начало.

$$(\text{cons } 'A \text{ } 'B) = (A . B) \qquad (\text{cons } 'A \text{ } '(B)) = (A B)$$

- list - всегда создает список, не является чистой, так как имеет произвольное количество аргументов.

(list 'A 'B 'C) = (A B C)

Практическая часть

4. Напишите результат вычисления выражений:

(list 'Fred 'and Wilma)		Wilma unbound
(cons 'Fred '(and Wilma))		(Fred and Wilma)
(list 'Fred '(and Wilma))		(Fred (and Wilma))
(cons 'Fred '(Wilma))		(Fred Wilma)
(cons Nil Nil)		(Nil)
(list Nil Nil)		(Nil Nil)
(cons T Nil)		(T)
(list T Nil)		(T Nil)

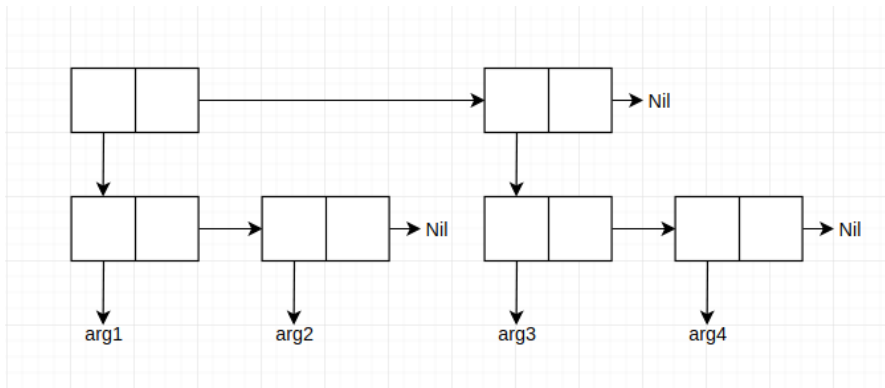
(cons Nil T)		(Nil . T)
(list Nil T)		(Nil T)
(list Nil)		(Nil)
(cons T (list Nil))		(T Nil)
(cons (T) Nil)		The function T is undefined
(list (T) Nil)		The function T is undefined
(list '(one two) '(free temp))		((one two) (free temp))
(cons '(one two) '(free temp))		((one two) free temp)

5. Написать функции

a. $(f\ ar1\ ar2\ ar3\ ar4) \rightarrow ((ar1\ ar2)\ (ar3\ ar4))$

`(defun f (ar1 ar2 ar3 ar4) (list (list ar1 ar2)(list ar3 ar4)))`

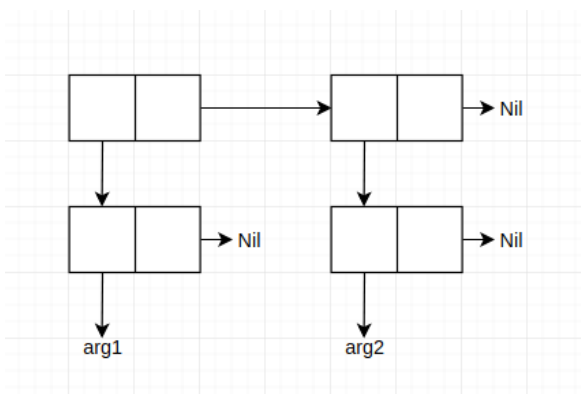
`(f 1 2 3 4) -> ((1 2)(3 4))`



b. $(f\ ar1\ ar2) \rightarrow ((ar1)\ (ar2))$

`(defun f (ar1 ar2) (list (list ar1)(list ar2)))`

`(f 1 2) -> ((1)(2))`



c. $(f\ ar1) \rightarrow (((ar1)))$

`(defun f (ar1) (list (list (list ar1))))`

$(f\ 1) \rightarrow (((1)))$

