



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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## ОТЧЕТ

по лабораторной работе № 1

по курсу «Функциональное и логическое программирование»

на тему: «Списки в Lisp. Использование стандартных функций»

Студент ИУ7-66Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Жаворонкова А. А.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Толпинская Н. Б.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Строганов Ю. В.  
(И. О. Фамилия)

2024 г.

# 1 Теоретические вопросы

## 1.1 Элементы языка: определение, синтаксис, представление в памяти

Вся информация (данные и программы) в Lisp представляется в виде символьных выражений — S-выражений. По определению:

S-выражение ::= <атом> | <точечная пара>.

Атомы:

- символы — синтаксически — набор литер (букв и цифр), начинающихся с буквы;
- специальные символы — T, Nil;
- самоопределимые атомы — натуральные числа, дробные числа, вещественные числа, строки — последовательность символов, заключенная в двойные апострофы.

Списки и точечные пары (структуры) строятся из унифицированных структур — бинарных узлов.

Точечная пара ::= (<атом>.<атом>) | (<атом>.<точечная пара>)  
| (<точечная пара>.<атом>)  
| (<точечная пара>.<точечная пара>)

Список ::= <пустой список> | <непустой список>  
<пустой список> ::= () | Nil ,  
<непустой список> ::= (<первый элемент>.<хвост>),  
<первый элемент> ::= <S-выражение>,  
<хвост> ::= <список>.

Любая структура (точечная пара или список) заключается в круглые скобки (A.B) — точечная пара, (A) — список из одного элемента, пустой список изображается как Nil или ().

Непустой список по определению может быть изображен: (A.(B.(C.(D())))). Допустимо изображение списка последовательностью атомов, разделенных пробелами — (A B C D).

Элементы списка могут, в свою очередь, быть списками (любой список заключается в круглые скобки), например — (A (B C) (D (E))). Таким образом,

синтаксически наличие скобок является признаком структуры — списка или точечной пары.

Любая непустая структура Lisp в памяти представляется списковой ячейкой, хранящей два указателя: на голову (первый элемент) и хвост — все остальное.

На рисунке 1.1 показано представление в памяти точечной пары (A.B) и списка из двух элементов — (A B).

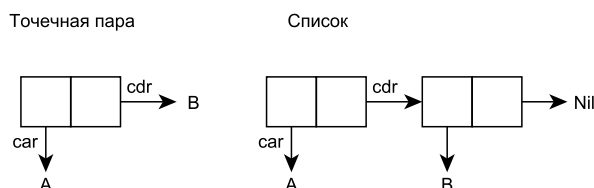


Рисунок 1.1 – Представление в памяти точечной пары и списка

## 1.2 Особенности языка Lisp. Структура программы.

### Символ апостроф

Особенности языка Lisp:

1. в Lisp используется символьная обработка;
2. программа и данные в Lisp представлены в виде списков (едины в своем физическом представлении).
3. Lisp является бестиповым языком;
4. память выделяется блоками. Lisp сам распределяет память.

Символ апостроф («'») — блокирует вычисление своего аргумента. В качестве своего значения выдает сам аргумент, не вычисляя его. Перед числами, T и Nil апостроф можно не ставить.

## 1.3 Базис языка Lisp. Ядро языка

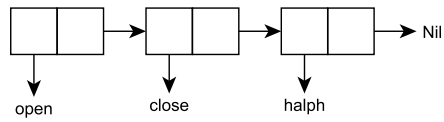
Базис — минимальный набор обозначений, к которым можно свести все правильные (вычисляемые) формулы системы. Базис Lisp образуют: атомы, структуры, базовые функции, базовые функционалы.

Ядро языка — совокупность базиса языка и наиболее часто используемых функций.

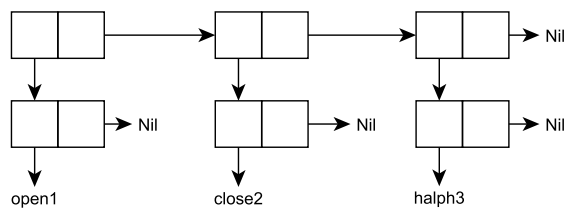
## 2 Практические задания

### 1. Представить следующие списки в виде списочных ячеек.

'(open close halph)



'((open1) (close2) (halph3))



'((one) for all (and (me (for you))))

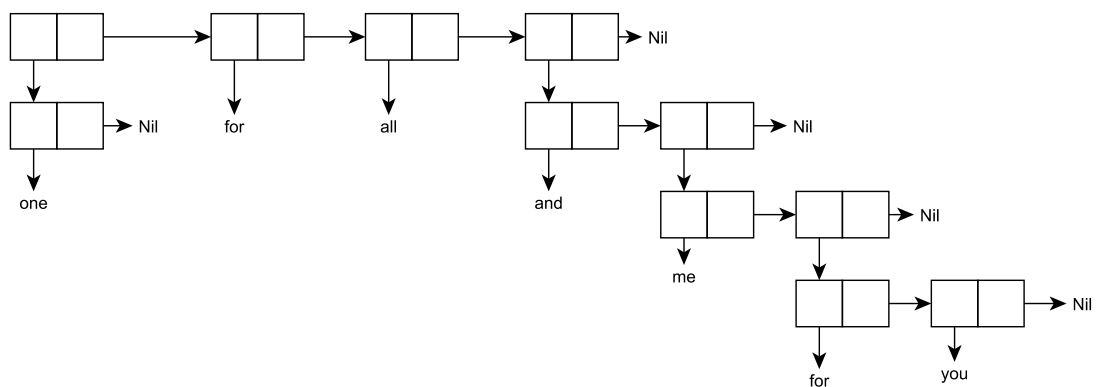
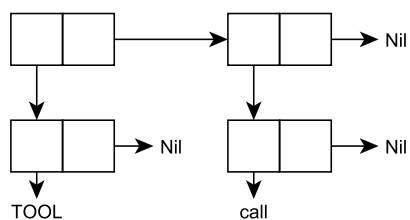
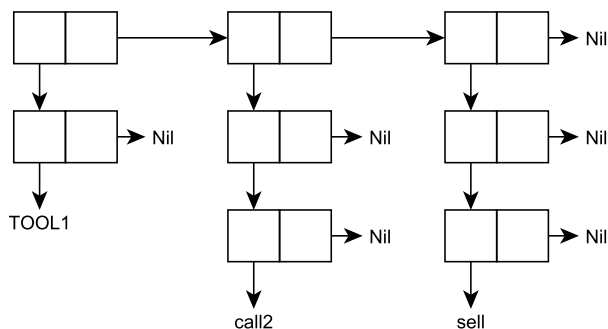


Рисунок 2.1 – Задание №1 (часть 1)

'((TOOL) (call))



'((TOOL1) ((call2)) ((sell)))



'(((TOOL) (call)) ((sell)))

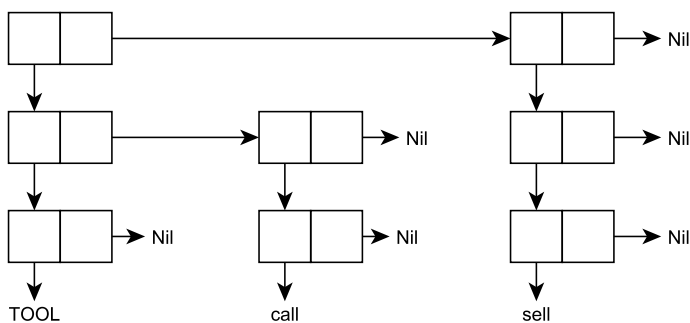


Рисунок 2.2 – Задание №1 (часть 2)

**2. Используя только функции CAR и CDR, написать выражения, возвращающие 1) второй; 2) третий; 3) четвертый элементы заданного списка.**

1) (CAR (CDR '(a b c d e)))

2) (CAR (CDR (CDR '(a b c d e))))

3) (CAR (CDR (CDR (CDR '(a b c d e)))))

**3. Что будет в результате вычисления выражений?**

a) (CAADR '((blue cube) (red pyramid)))  
(CAR (CAR (CDR '((blue cube) (red pyramid))))))  
(CAR (CAR '(red pyramid)))  
(CAR '(red pyramid))  
red

ОТВЕТ: red

b) (CDAR '((abc) (def) (ghi)))  
(CDR (CAR '((abc) (def) (ghi))))  
(CDR '(abc))  
Nil

ОТВЕТ: Nil

c) (CADR '((abc) (def) (ghi)))  
(CAR (CDR '((abc) (def) (ghi))))  
(CAR '((def) (ghi)))  
(def)

ОТВЕТ: (def)

d) (CADDR '((abc) (def) (ghi)))  
(CAR (CDR (CDR '((abc) (def) (ghi))))))  
(CAR (CDR '((def) (ghi))))  
(CAR '((ghi)))  
(ghi)

ОТВЕТ: (ghi)

**4. Напишите результат вычисления выражений и объясните как он получен.**

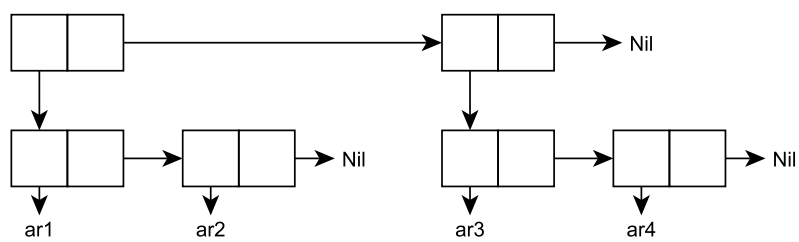
1. (list 'Fred 'and 'Wilma)  $\rightarrow$  (Fred and Wilma)
2. (list 'Fred '(and Wilma))  $\rightarrow$  (Fred (and Wilma))
3. (cons Nil Nil)  $\rightarrow$  (Nil)
4. (cons T Nil)  $\rightarrow$  (T)
5. (cons Nil T)  $\rightarrow$  (Nil.T)

6.  $(\text{list Nil}) \rightarrow (\text{Nil})$
7.  $(\text{cons '}(T) \text{ Nil}) \rightarrow ((T))$
8.  $(\text{list '}(one\ two) \text{ '}(free\ temp))) \rightarrow ((one\ two) (free\ temp))$
9.  $(\text{cons 'Fred '}(and\ Wilma)) \rightarrow (\text{Fred and Wilma})$
10.  $(\text{cons 'Fred '}(Wilma)) \rightarrow (\text{Fred Wilma})$
11.  $(\text{list Nil Nil}) \rightarrow (\text{Nil Nil})$
12.  $(\text{list T Nil}) \rightarrow (\text{T Nil})$
13.  $(\text{list Nil T}) \rightarrow (\text{Nil T})$
14.  $(\text{cons T (list Nil)}) \rightarrow (\text{T Nil})$
15.  $(\text{list '}(T) \text{ Nil}) \rightarrow ((T) \text{ Nil})$
16.  $(\text{cons '}(one\ two) \text{ '}(free\ temp))) \rightarrow ((one\ two) free\ temp)$

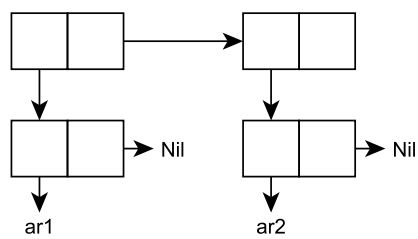
**5. Написать лямбда-выражение и соответствующую функцию.  
Представить результаты в виде списочных ячеек.**

1. Написать функцию  $(f\ ar1\ ar2\ ar3\ ar4)$ , возвращающую:  $((ar1\ ar2) (ar3\ ar4))$ .  
 $(\text{defun } f(ar1\ ar2\ ar3\ ar4) (\text{list (list } ar1\ ar2) (\text{list } ar3\ ar4)))$
2. Написать функцию  $(f\ ar1\ ar2)$ , возвращающую:  $((ar1) (ar2))$ .  
 $(\text{defun } f(ar1\ ar2) (\text{list (list } ar1) (\text{list } ar2))))$
3. Написать функцию  $(f\ ar1)$ , возвращающую:  $((ar1))$ .  
 $(\text{defun } f(ar1) (\text{list (list (list } ar1))))$

((ar1 ar2) (ar3 ar4))



((ar1) (ar2))



((ar1)))

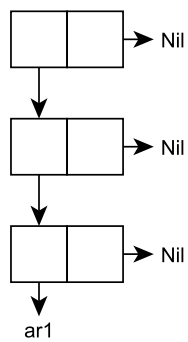


Рисунок 2.3 – Задание №5