



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»

Отчет по лабораторной работе №1 по дисциплине "Функциональное и логическое программирование"

Тема Списки в Lisp. Использование стандартных функций.

Студент Егорова П.А.

Группа ИУ7-64Б

Преподаватели Толпинская Н.Б., Строганов Ю.В.

Москва — 2023 г.

Содержание

1	Цель работы	3
2	Теоретические вопросы	4
2.1	Элементы языка: определение, синтаксис, представление в памяти	4
2.2	Особенности языка Lisp. Структура программы. Символ апостроф	5
2.3	Базис языка Lisp. Ядро языка	5
3	Практические задания	7
3.1	Вывод	11
	Список использованных источников	11

1 Цель работы

Цель работы: приобрести навыки использования списков и стандартных функций Lisp.

2 Теоретические вопросы

2.1 Элементы языка: определение, синтаксис, представление в памяти

Базис Lisp образуют: атомы, структуры, базовые функции, базовые функционалы [1].

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

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

Атомы:

- *символы* — синтаксически — набор литер (букв и цифр), начинающийся с буквы;
- *специальные символы* $\{T, Nil\}$ обозначают логические константы;
- *самоопределимые атомы* — натуральные, дробные, вещественные числа и строки.

Более сложные данные — списки и точечные пары (структуры).

Точечные пары ::= (<атом>.<атом>) | (<атом>.<точечная пара>) | (<точечная пара>.<атом>) | (<точечная пара>.<точечная пара>);

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

Синтаксически любая структура заключается в круглые скобки:

- (A . B) — точечная пара;
- (A) — список из одного элемента;

- Nil или () – пустой список;
- (A . (B . (C . (D ()))) или (A B C D) – непустой список;
- элементы списка могут являться списками: ((A)(B)(CD)).

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

2.2 Особенности языка Lisp. Структура программы. Символ апостроф

Особенности языка Lisp:

- символьная обработка данных;
- любая программа может интерпретироваться как функция с одним или несколькими аргументами;
- автоматизированное динамическое распределение памяти, которая выделяется блоками;
- бестиповый язык;
- программа может быть представлена как данные, то есть программа может изменять саму себя.

Символ апостроф — сокращенное обозначение функции quote, блокирующей вычисление своего аргумента.

2.3 Базис языка Lisp. Ядро языка

Базис языка — минимальный набор конструкций и структур данных, с помощью которого можно написать любую программу.

Базис Lisp образуют:

- атомы;
- структуры;
- базовые функции;
- базовые функционалы.

Ядро – основные действия, которые наиболее часто используются. Ядро шире, чем базис.

3 Практические задания

Решение задания №1 представлено на рисунках 3.1 – 3.6.

' (open close halph)

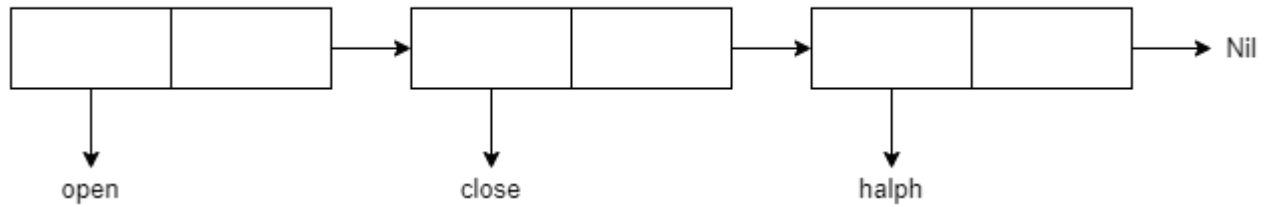


Рисунок 3.1 – Задание 1, часть 1

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

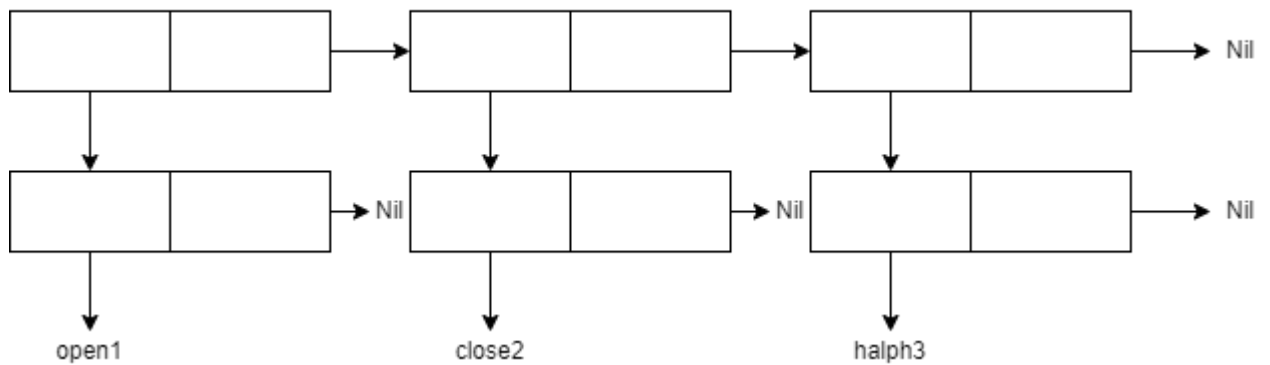


Рисунок 3.2 – Задание 1, часть 2

' ((TOOL) (call))

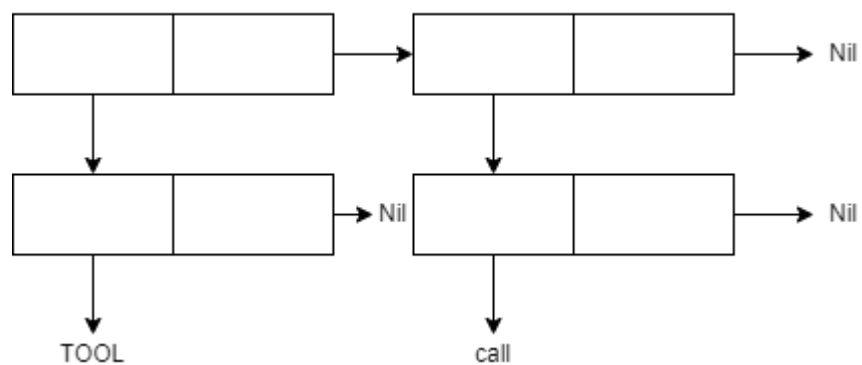


Рисунок 3.3 – Задание 1, часть 3

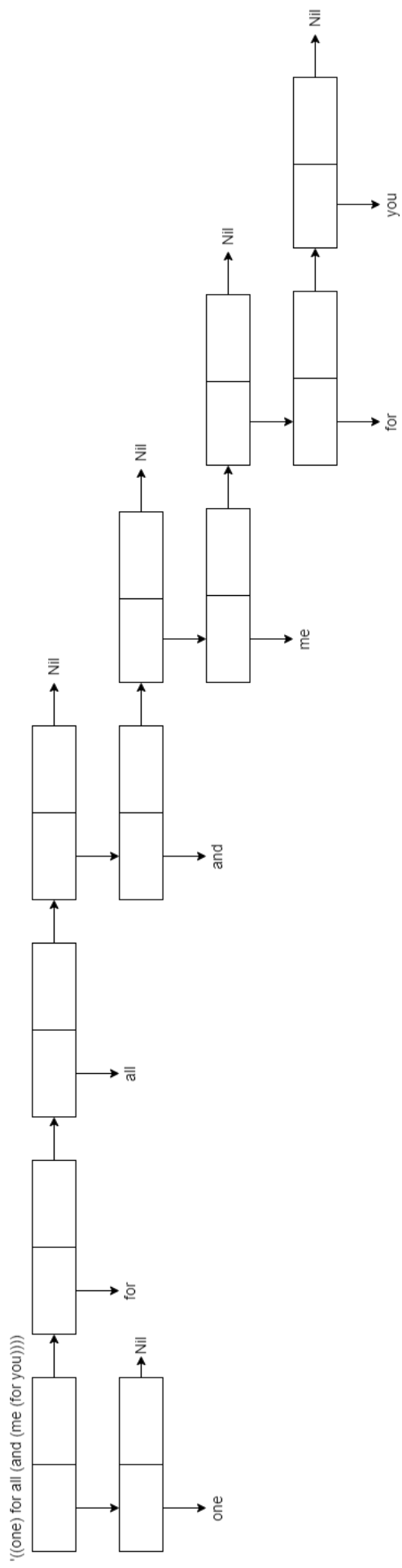


Рисунок 3.4 – Задание 1, часть 4

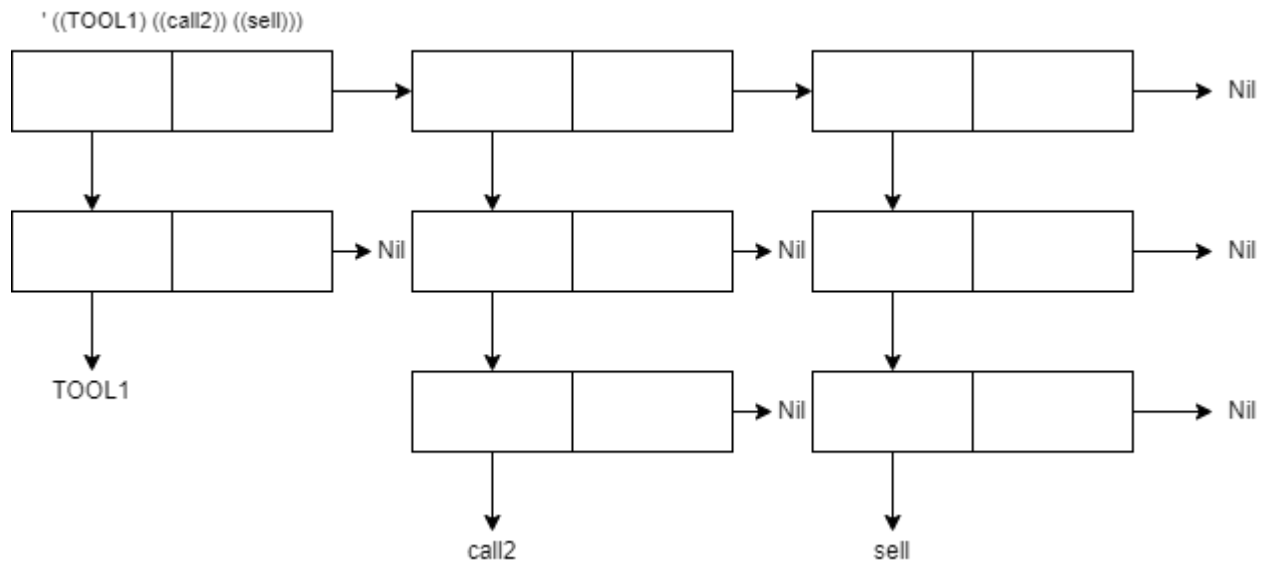


Рисунок 3.5 – Задание 1, часть 5

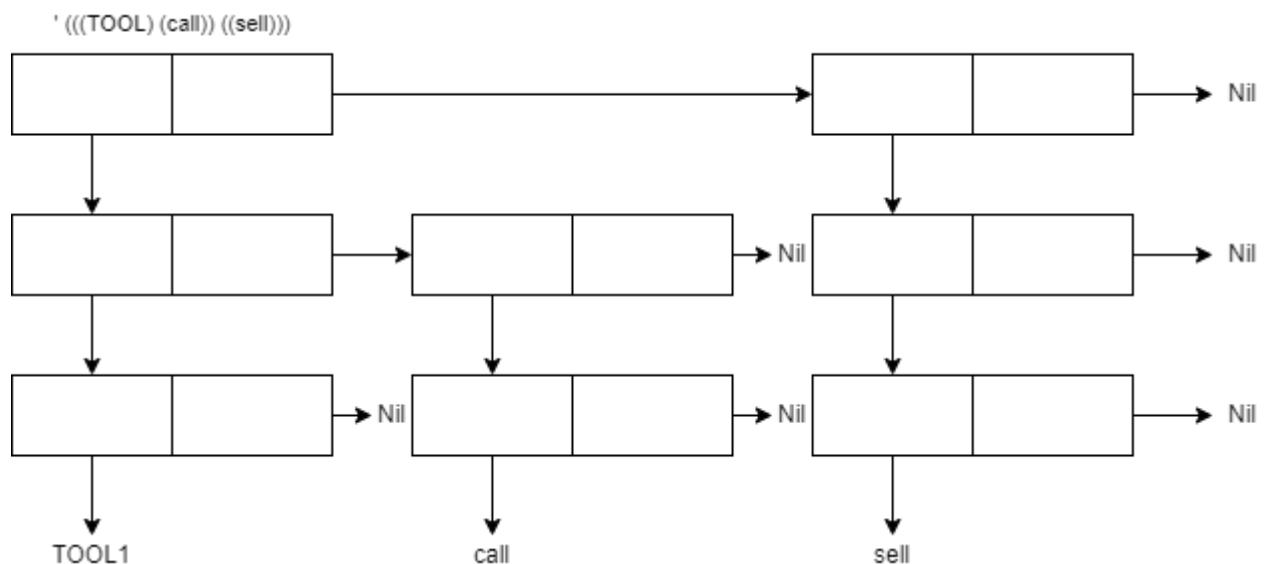


Рисунок 3.6 – Задание 1, часть 6

Решение задания №2 представлено на рисунке ??.

Задание 2

Используя только функции CAR и CDR, написать выражения, возвращающие... пусть будет список (a b c d)

2-й элемент: (car '(cdr '(a b c d))) - сначала cdr оставит (b c d), а потом car вернет первый атом, то есть b

3-й элемент: (car '(cdr '(cdr '(a b c d)))) - сначала cdr оставит (b c d), потом (c d), а потом car вернет первый атом, то есть c

4-й элемент: (car '(cdr '(cdr '(cdr '(a b c d)))))

Рисунок 3.7 – Задание 2

Решение задания №3 представлено на рисунке ??.

Задание 3

- a) (CAADR ' ((blue cube) (red pyramid))) -> red
- b) (CDAR ' ((abc) (def) (ghi))) -> Nil
- c) (CADR ' ((abc) (def) (ghi))) -> (def)
- d) (CADDR ' ((abc) (def) (ghi))) -> (ghi)

Рисунок 3.8 – Задание 3

Решение задания №4 представлено на рисунке 3.9.

Задание 4

(list 'Fred 'and 'Wilma) -> (Fred and Wilma)	(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) -> ((T))	(list '(T) Nil) -> ((T) Nil)
(list ' (one two) ' (free temp)) -> ((one two) (free temp))	(cons '(one two) '(free temp)) -> ((one two) free tmp)

Рисунок 3.9 – Задание 4

Решение задания №5.

Функция (f ar1 ar2 ar3 ar4), возвращающая ((ar1 ar2) (ar3 ar4)).

```
1 (defun f1 (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
2 (f1 9 5 3 2) => ((9 5) (3 2))
3 ;
4 (defun f(ar1 ar2 ar3 ar4) (cons (cons ar1 (cons ar2 nil)) (cons
5   (cons ar3 (cons ar4 nil)) nil)))
6 ;; lambda
6 (lambda (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
7 ((lambda (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4))) 6
7 1 2) => ((6 7) (1 2))
```

Функция (f ar1 ar2), возвращающая ((ar1) (ar2))

```
1 (defun f2 (ar1 ar2) (list (list ar1) (list ar2)))
2 (f2 1 2) => ((1) (2))
3 ;
4 (defun f(ar1 ar2) (cons (cons ar1 nil) (cons (cons ar2 nil) nil)))
5 ;; lambda
```

```

6 (lambda (ar1 ar2) (list (list ar1) (list ar2)))
7 ((lambda (ar1 ar2) (list (list ar1) (list ar2))) 1 2) => ((1) (2))

```

Функция (f ar1), возвращающая (((ar1)))

```

1 (defun f3 (ar1) (list (list (list ar1))))
2 (f3 9) => (((9)))
3 ;
4 (defun f(ar1) (cons (cons (cons ar1 nil) nil) nil))
5 ;; lambda
6 (lambda (ar1) (list (list (list ar1))))
7 ((lambda (ar1) (list (list (list ar1)))) 9) => (((9)))

```

Результаты в виде списочных ячеек представлены на рисунке 3.10.

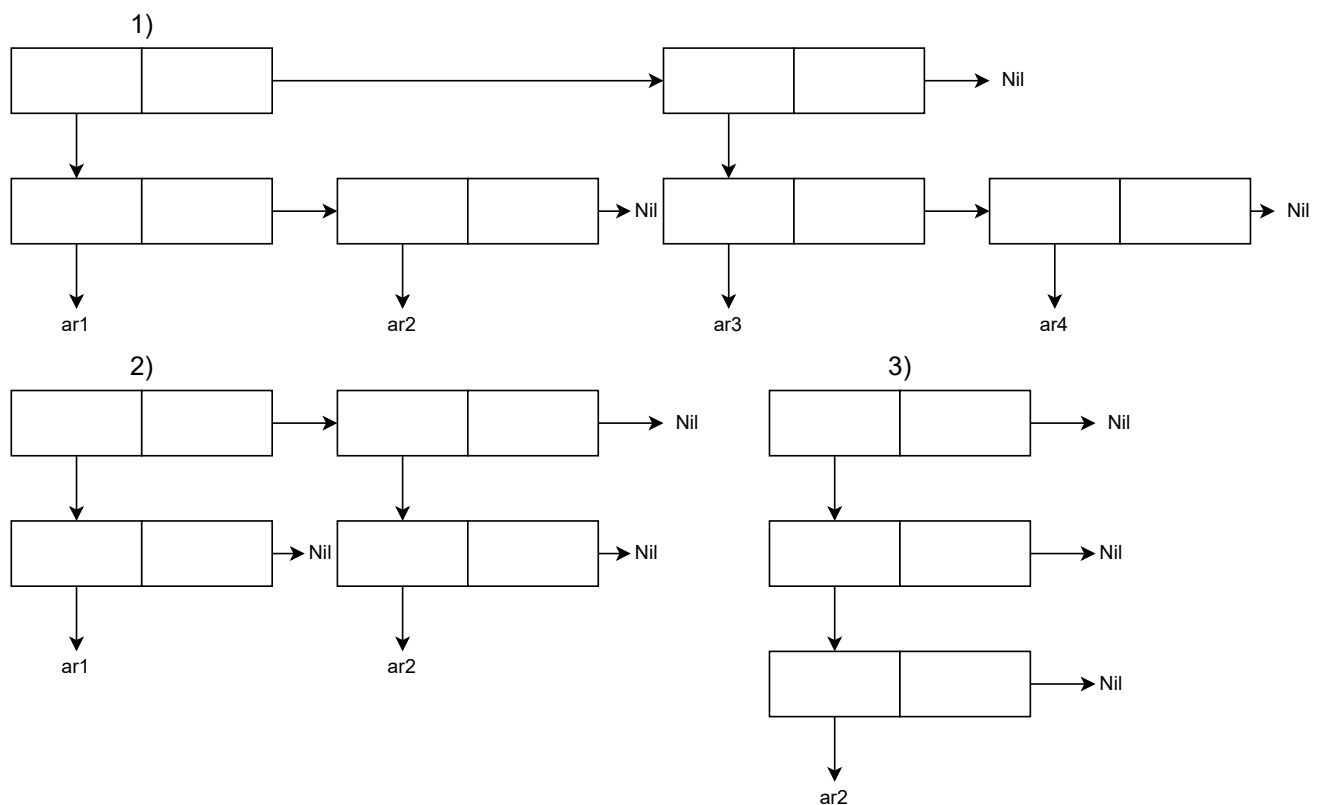


Рисунок 3.10 – Задание 5

3.1 Вывод

В ходе выполнения лабораторной работы была достигнута поставленная ранее цель: были приобретены навыки использования списков и стандартных функций Lisp.

Список использованных источников

1. Лабораторная работа №1. Методические указания. Списки в Lispe. Использование стандартных функций.