



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»

Отчёт по лабораторной работе № 1 по курсу «Функциональное и логическое программирование»

Тема Списки в Lisre. Использование стандартных функций

Студент Волков Г. В.

Группа ИУ7-61Б

Оценка (баллы) _____

Преподаватель Толпинская Н. Б.

1 Теоретические вопросы

Вопрос 1. Элементы языка: определение, синтаксис, представление в памяти

Элементы языка и их определение

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

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

Атомами являются: символы — набор литер, специальные символы — T, Nil, самоопределимые атомы — числа, строки.

Более сложные данные — списки и точечные пары (структуры) строятся из унифицированных структур — бинарных узлов.

Точечная пара — структура данных, состоящая из двух символьных выражений, разделенных точкой. По определению:

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

Список — это структура, которая может быть пустой и непустой. Если непустой, то состоит из двух элементов: голова — любой элемент, хвост — список. По определению:

Список ::= <пустой список> | <непустой список>), где

<пустой список> ::= () | Nil,

<непустой список> ::= (<первый элемент>, <хвост>),

<первый элемент> ::= (S-выражение),

<хвост> ::= <список>

Синтаксис элементов языка и их представление в памяти

Элементарной синтаксической конструкцией языка является атом. Атом начинающийся с цифры или унарного знака считается числовым, иначе символьным.

Любая структура (точечная пара или список) заключается в (). Пример:

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

Пустой список изображается как Nil или (). Непустой список может быть записан как (A . (B . (C . Nil))) или (A B C). Элементами списка могут быть атомы или другие структуры. Любая структура представляется в памяти с помощью бинарных узлов (списковых ячеек), которые хранят два указателя: на голову и хвост.

Вопрос 2. Особенности языка Lisp. Структура программы. Символ апостроф

Lisp — интерпретируемый символьный язык программирования, т. е. язык предназначен для символьных вычислений и преобразований. В основе языка лежит идея λ -исчисления.

Программа и данные в Lisp представлены списками. Синтаксического различия между программой и данными нет. Это позволяет использовать программу как данные и заставить её менять саму себя. По умолчанию список считается вычислимой формой, в которой первый элемент — название функции, остальные элементы — аргументы функции.

Поскольку программа и данные представлены списками они неразличимы. Для их различия существует функция `quote`, блокирующая вычисление. Символ ' является её сокращенным обозначением.

Вопрос 3. Базис языка Lisp. Ядро языка

Базис — это минимально необходимый набор конструкций, к которому могут быть сведены другие конструкции языка.

Базис Lisp образуют:

- 1) атомы;
- 2) структуры;
- 3) базовые функции;
- 4) функционалы.

2 Практические задания

Задание 1

Представить следующие списки в виде списочных ячеек:

- 1) `'(open close halph)`
- 2) `'((open1) (close2) (halph3))`
- 3) `'((one) for all (and (me (for you))))`
- 4) `'((TOOL) (call))`
- 5) `'((TOOL1) ((call2)) ((sell)))`
- 6) `'(((TOOL) (call)) ((sell)))`

Решение:

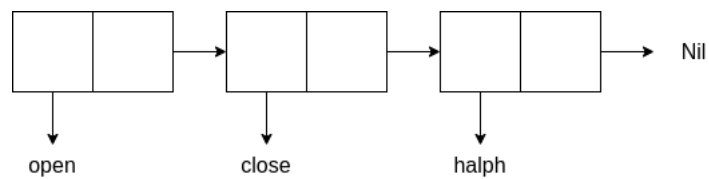


Рисунок 2.1 – `'(open close halph)`

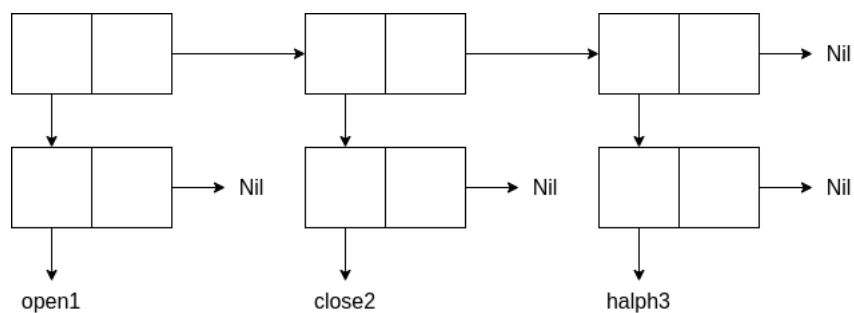


Рисунок 2.2 – `'((open1) (close2) (halph3))`

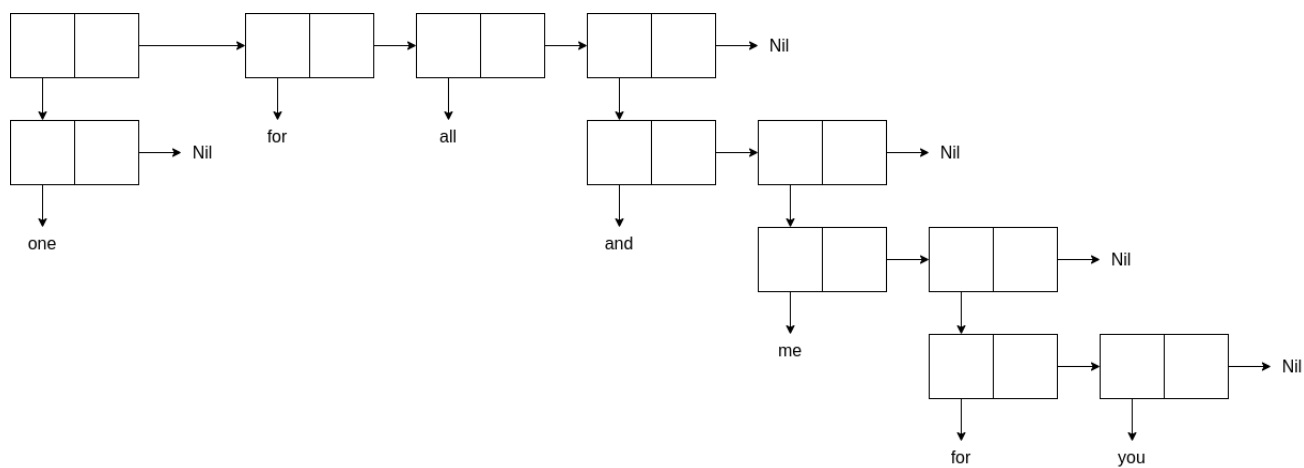


Рисунок 2.3 – '((one) for all (and (me (for you))))

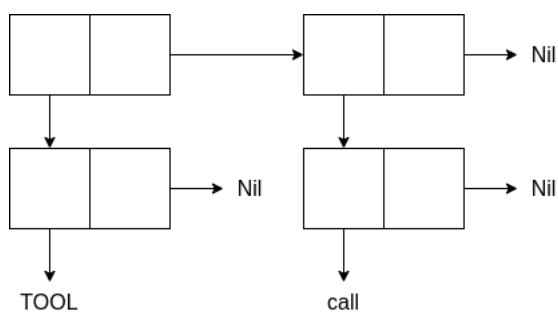


Рисунок 2.4 – '((TOOL) (call))

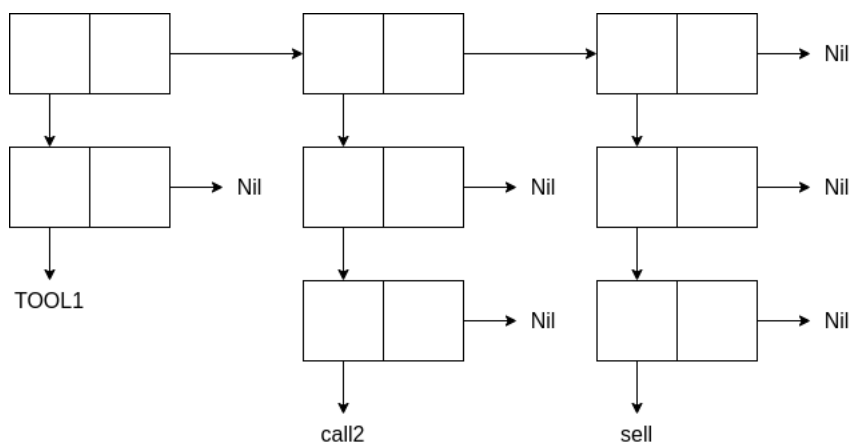


Рисунок 2.5 – '((TOOL1) ((call2)) ((sell)))

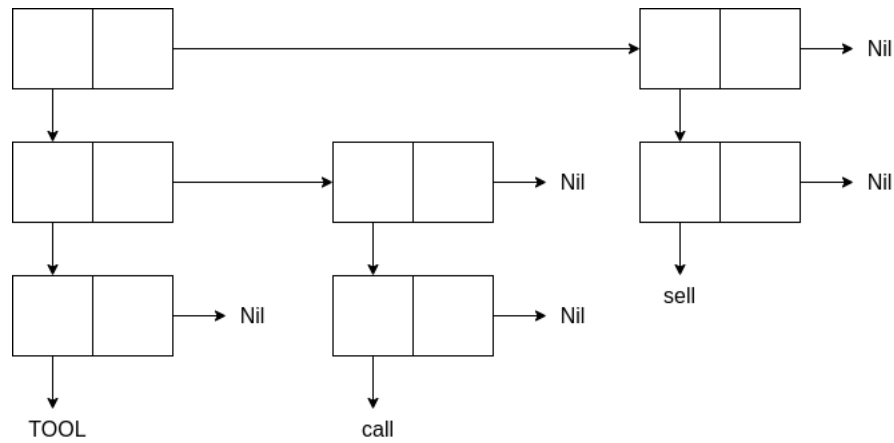


Рисунок 2.6 – '(((TOOL) (call)) ((sell)))

Задание 2

Используя только функции CAR и CDR, написать выражения, возвращающие:

- 1) второй;
- 2) третий;
- 3) четвёртый;

элементы заданного списка.

Решение:

Листинг 2.1 – Задание 2

```
1 (car (cdr '(1 2 3 4 5))) ; 2
2 (car (cdr (cdr '(1 2 3 4 5)))) ; 3
3 (car (cdr (cdr (cdr '(1 2 3 4 5))))) ; 4
```

Задание 3

Что будет в результате вычисления выражений?

Листинг 2.2 – Задание 3

```
1 (CAADR '((blue cube) (red pyramid))) ; -> red
```

```
2 (CDAR '((abc) (def) (ghi))) ; -> Nil
3 (CADR '((abc) (def) (ghi))) ; -> (def)
4 (CADDR '((abc) (def) (ghi))) ; -> (ghi)
```

Задание 4

Напишите результат вычисления выражений и объясните как он получен:

Листинг 2.3 – Задание 4

```
1 (list 'Fred 'and 'Wilma) ; -> (fred and wilma)
2 (list 'Fred '(and Wilma)) ; -> (fred (and wilma))
3 (cons Nil Nil) ; -> (Nil)
4 (cons T Nil) ; -> T
5 (cons Nil T) ; -> (Nil . T)
6 (list Nil) ; -> (Nil)
7 (cons '(T) Nil) ; -> ((T))
8 (list '(one two) '(free temp)) ; -> ((one two) (free temp))
9 (cons 'Fred '(and Wilma)) ; -> (Fred and Wilma)
10 (cons 'Fred '(Wilma)) ; -> (Fred Wilma)
11 (list Nil Nil) ; -> (Nil Nil)
12 (list T Nil) ; -> (T Nil)
13 (list Nil T) ; -> (Nil T)
14 (cons T (list Nil)) ; -> (T Nil)
15 (list '(T) Nil) ; -> ((T) Nil)
16 (cons '(one two) '(free temp)) ; -> ((one two) free temp)
```

Задание 5

Написать λ -выражение и соответствующую функцию и представить результаты в виде списочных ячеек.

- написать функцию (f ar1 ar2 ar3 ar4), возвращающую список ((ar1 ar2) (ar3 ar4));
- написать функцию (f ar1 ar2), возвращающую список ((ar1) (ar2));
- (f ar1), возвращающую список ((ar1)).

Листинг 2.4 – Задание 5

```

1 ; 1)
2 (lambda (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
3 (defun f1 (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
4 ; 2)
5 (lambda (ar1 ar2) (list (list ar1) (list ar2)))
6 (defun f2 (ar1 ar2) (list (list ar1) (list ar2)))
7 ; 3)
8 (lambda (ar1) (list (list (list ar1))))
9 (defun f3 (ar1) (list (list (list ar1))))

```

Результаты в виде списочных ячеек представлены на рисунках

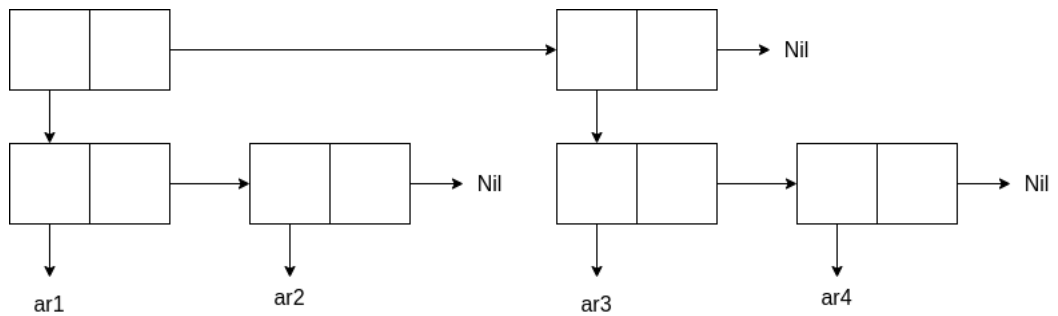


Рисунок 2.7 – ((ar1 ar2) (ar3 ar4))

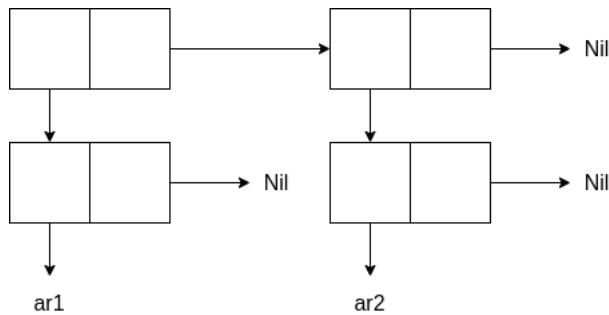


Рисунок 2.8 – ((ar1) (ar2))

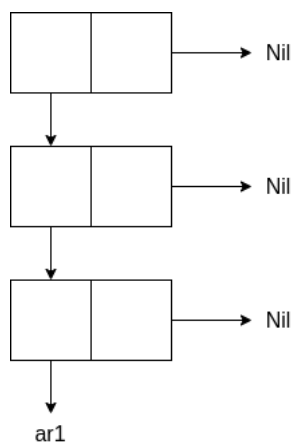


Рисунок 2.9 – (((ar1)))