



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»

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

Тема Представление списков в виде списочных ячеек

Студент Богаченко А. Е.

Группа ИУ7-65Б

Оценка (баллы) _____

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

Задание 1

Представить следующие списки в виде списочных ячеек:

1. `'(open close halp)`
2. `'((open1) (close2) (halp3))`
3. `'((one) for all (and (me (for you))))`
4. `'((TOOL) (call))`
5. `'((TOOL1) ((call2)) ((sell)))`
6. `'(((TOOL) (call)) ((sell)))`

Решение оформлено на тетрадном листе.

Задание 2

Используя только функции `car` и `cdr`, написать выражения, возвращающие второй, третий, четвёртый элементы заданного списка:

Листинг 1 – Задание 1

```
1 (defun list-nth (l idx)
2   (case idx
3     (2 (cadr l))
4     (3 (caddr l))
5     (4 (cadddr l))))
```

Задание 3

Что будет в результате выполнения выражений?

Листинг 2 – Задание 2

```
1 (caddr '((blue cube)(red pyramid))) ; -> red
2 (cdar '((abc)(def)(ghi))) ; -> Nil
3 (cadr '((abc)(def)(ghi))) ; -> (def)
4 (cadddr '((abc)(def)(ghi))) ; -> (ghi)
```

Задание 4

Напишите результат вычисления выражений:

Листинг 3 – Задание 3

```
1 (list 'Fred 'and Wilma) ; -> UNBOUND VARIABLE
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) ; -> UNDEFINED FUNCTION
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) ; -> UNDEFINED FUNCITON
16 (cons '(one two) '(free temp)) ; -> ((one two) free temp)
```

Задание 5

Написать:

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

результаты представить в виде списочных ячеек.

Листинг 4 – Задание 4

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

Представление полученных списков в виде списочных ячеек оформлено на тетрадном листе.

Контрольные вопросы

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

Элементы языка

Элементами языка Lisp являются атомы и точечные пары (структуры).
К атомам относятся:

- символы (идентификаторы) – набор литер, начинающихся с буквы;
- специальные символы для обозначения логических констант T, Nil;
- самоопределимые атомы – натуральные числа, дробные числа, вещественные числа, строки (последовательность символов, заключённых в двойные апострофы).

Синтаксис элементов языка и их представление в памяти

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>

Список – частный случай S-выражения.

Синтаксически любая структура (точечная пара или список) заключается в ():

(A . B) – точечная пара

(A) – список из одного элемента

Пустой список изображается как Nil или ()

Непустой список может быть изображён: (A. (B . (C ()))) или (A B C)

Элементы списка могут являться списками: ((A) (B) (C))

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

2. Особенности языка LISP. Структура программы. Символ апостроф

Структура программы

Lisp - язык символьной обработки. В Lisp программа и данные представлены списками. По умолчанию список считается вычислимой формой, в которой 1 элемент - название функции, остальные элементы - аргументы функции.

Особенности языка

Т.к. и программа и данные представлены списками, то их нужно как-то различать. Для этого была создана функция `quote`, а `'` - ее сокращенное обозначение. `quote` - функция, блокирующая вычисление.

Символ апостроф

Символ `'` – функциональная блокировка, эквивалентен функции `quote`. Блокирует вычисление выражения. Таким образом, выражение воспринимается интерпретатором как данные.

3. Базис языка LISP. Ядро языка

Базис языка представлен:

- структурами, атомами;
- функциями:

`atom`, `eq`, `cons`, `car`, `cdr`,

`cond`, `quote`, `lambda`, `eval`, `label`.