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 (ИУ7)**

О Т Ч Е Т

по лабораторной работе № 0 1

Дисциплина: Функциональное и логическое программирование

Студент ИУ7И-62Б _____ Во Д. Х. Ф
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Толпинская Н. Б.
_____ Строганов Ю. В.
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

Теоретические вопросы

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

Элементы языка:

Атом:

- символы (идентификаторы) — синтаксически — набор литер (букв и цифр), начинающихся с буквы;
- специальные символы — {T, Nil} (используются для обозначения логических констант);
- самоопределимые атомы — натуральные числа, дробные числа (например 2/3), вещественные числа, строки — последовательность символов, заключённых в двойные апострофы (например "abc").

Точечная пара ::= (<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>

Синтаксис элемента языка и их представление в памяти

Синтаксически любая структура (точечная пара или список) заключается в круглые скобки.

(A . B) – точечная пара

(A) – список из оного 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 быть списками (любой список заключается в круглые скобки), например — (A (B C) (D (E))).

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

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

Вся информация (данные и программы) в Lisp представляется в виде символьных выражений — S-выражений.
По определение S-выражение ::= <атом> | <точечная пара>

Функция *quote* предохраняет свой единственный аргумент от вычисления (блокирует вычисление). Использование апострофа ' — просто сокращённое обозначение функции *quote*.

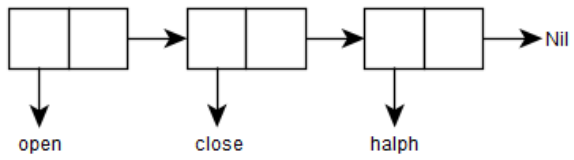
3. Базис языка Lisp. Ядро языка.

Базис Lisp образуют: атомы, структуру из простейших бинарных узлов, базовые функции (atom, eq, cons, car, cdr) и базовые функционалы (quote, lambda, eval).

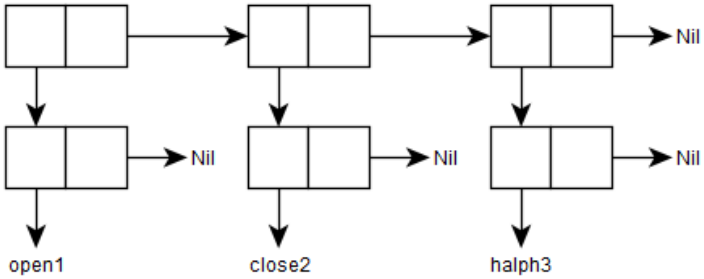
Функционнальное программирование ориентировано на символьную обработку данных. Предпологается, что любую информацию можно свести к символьной. Слово “символ” здесь близко к понятию “индентификатор”

Практические задание

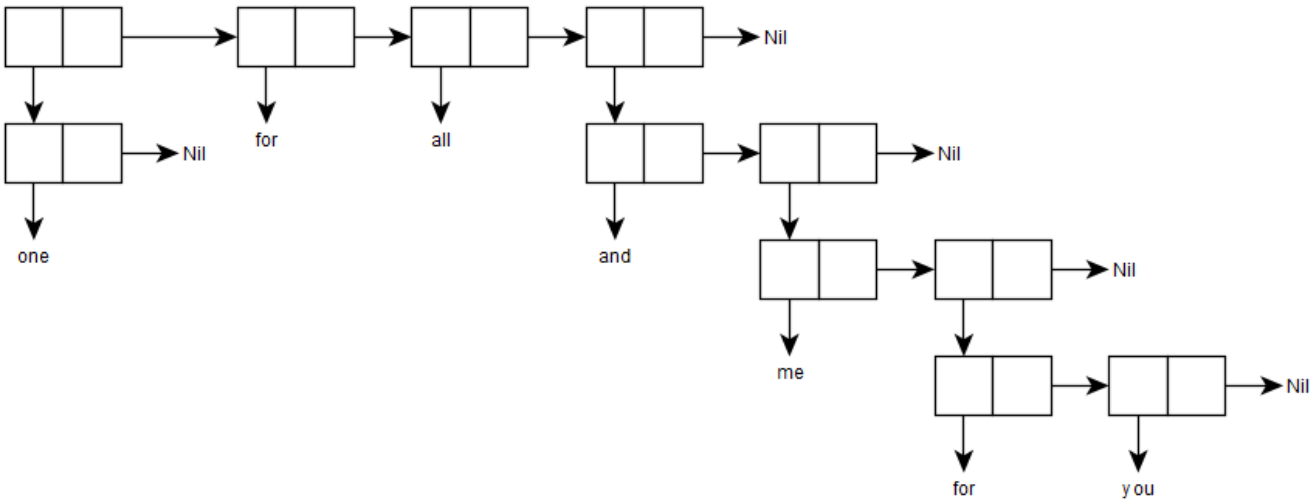
Задание 1. Представить следующие списки в виде списочных ячеек:
1. ‘(open close halph)



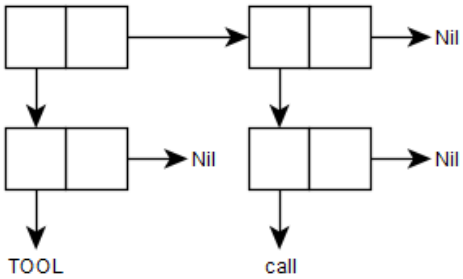
2. ‘((open1) (close2) (halph3))



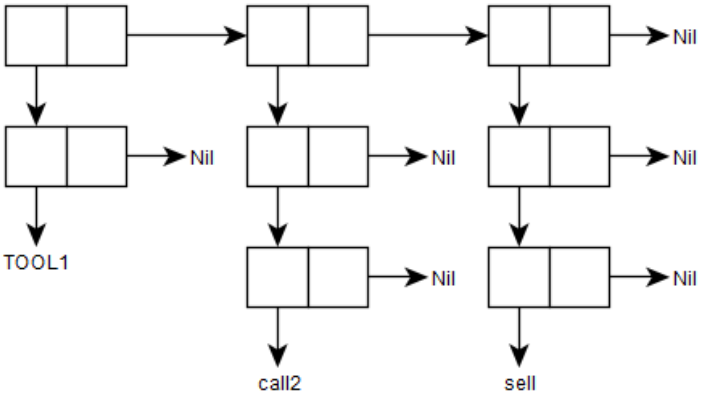
3. ‘((one) for all (and(me(for you))))



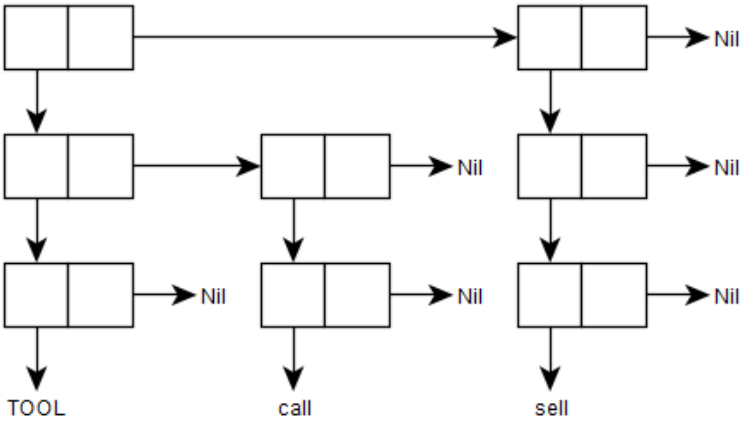
4. ‘((TOOL) (call))



5. ‘((TOOL1) ((call2)) ((sell)))



6. ‘(((TOOL) (call)) ((sell)))



Задание 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)))))

Задание 3. Что будет в результате вычисления выражений?

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)

Задание 4. Напишите результат вычисления выражения

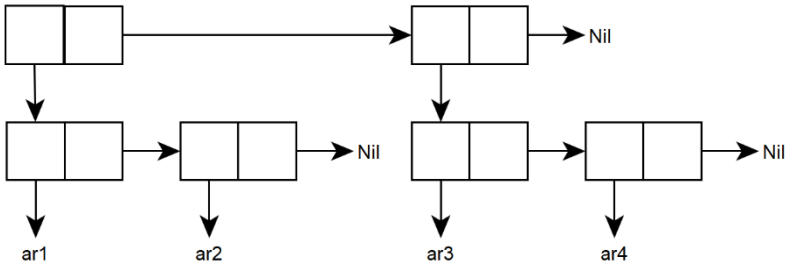
1. (list ‘Fred ‘and ‘Wilma)
;(Fred and Wilma)
2. (list ‘Fred ‘(and Wilma))
; (Fred (and Wilma))
3. (cons Nil Nil)
; (Nil)
4. (cons T Nil)
; (T)
5. (cons Nil T)
; (Nil . T)
6. (list Nil)
; (Nil)
7. (cons ‘(T) Nil)
; ((T))
8. (list ‘(one two) ‘(free temp))
; ((one two) (free temp))
9. (cons ‘Fred ‘(and 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 лямбда-выражение и соответствующую функцию:

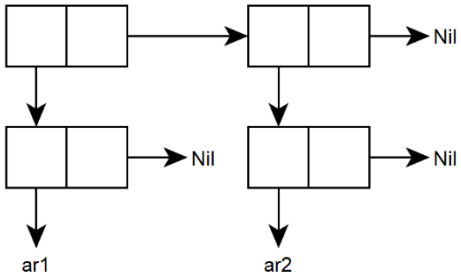
- Написать функцию (f ar1 ar2 ar3 ar4), возвращающую список: ((ar1 ar2) (ar3 ar4)

- Написать функцию (f ar1 ar2), возвращающую ((ar1) (ar2))
- Написать функцию (f ar1), возвращающую (((ar1)))
- Представить результаты в виде списочных ячеек.

1. (defun f (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))
   (lambda (ar1 ar2 ar3 ar4) (list (list ar1 ar2) (list ar3 ar4)))



2. (defun f (ar1 ar2) (list (list ar1) (list ar2)))
   (lambda (ar1 ar2) (list (list ar1) (list ar2)))



3. (defun f (ar1) (list (list (list ar1))))
   (lambda (ar1) (list (list (list ar1))))