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

Название «Списки в Lisp. Использование стандартных функций»

Дисциплина «Функциональное и логическое программирование»

Студент ИУ7-65Б _____ Бугаенко А.П.
(подпись, дата) (Фамилия И.О.)

Преподователь _____ Толпинская Н.Б.
(подпись, дата) (Фамилия И.О.)

Москва, 2022

1 Цели и задачи работы

Цель работы — приобрести навыки использования списков и стандартных функций Lisp. Задачи работы — изучить способ использования списков для фиксации информации, внутреннее представление одноуровневых и структурированных списков, методы их обработки с использованием базовых функций Lisp.

2 Теоретические вопросы

2.1 Элементы языка: определение, синтаксис, представление в памяти

Атом — элементарные данные. Атомами являются:

— символы (идентификаторы) — набор литер (литера — буква или цифра), начинающихся с буквы;

— специальные символы — T и Nil;

— самоопределяемые атомы — натуральные числа, дробные числа, вещественные числа, строки (последовательность символов, заключённая в двойные апрстрофы).

Точечная пара — единая универсальная базовая структура для конструирования сложных символьных выражений. Представляет из себя сложную структуру, состоящую из двух атомов. В памяти представляется бинарным узлом, левая и правая части которого равноправны и хранят указатели на эти атомы. Вместо атомов могут быть использованы более сложные выражения. Точечная пара ::= (<атом>.<атом>) | (<атом>.<точечная пара>) | (<точечная пара>.<атом>) | (<точечная пара>.<точечная пара>)

S-выражение (символьное выражение) - это или атом или заключённая в скобки пара из двух S-выражений, разделённых точкой. Все сложные символьные выражения являются S-выражениями, а в памяти представляются структурами из одинаково устроенных блоков - бинарных узлов, содержащих указатели на S-выражения. Каждый бинарный узел соответствует минимальному блоку памяти из двух указателей.

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

Список — рекурсивная структура, которая может быть описана следующими БНФ (формулами Бэкуса-Наура):

список ::= (последовательность_элементов) | пустой список

последовательность_элементов ::= элемент | элемент последовательность_элементов

элемент ::= атом | список

пустой список ::= () | NIL

Пробелы используются в качестве разделителей элементов списка.

2.2 Особенности языка Lisp. Структура программы. Символ апостроф

Особенности языка Lisp включают в себя:

— возможность обработки символьной информации (сложных символьных структур);

— единая синтаксическая форма представления данных и программы;

— ссылочная организация памяти;

— сборка мусора (автоматизация повторного использования памяти);

— смешанная схема трансляции — часть программы может быть обработана как данные и наоборот;

— наличие макросов;

— наличие интерпретатора.

Структура программы — программа и данные в lisp представлены списками. По умолчанию список считается вычислимой формой в которой 1 элемент - название функции, остальные элементы - аргументы функции.

Символ апостроф — ’ — сокращенное обозначение функции quote, которая блокирует вычисление.

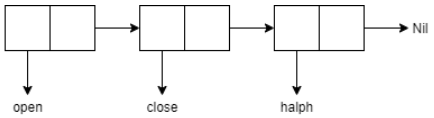
2.3 Базис языка Lisp. Ядро языка

Базис языка Lisp — атомы и структуры (представляющиеся бинарными узлами), базовые функции и функционалы: встроенные — примитивные функции (atom, eq, cons, car, cdr); специальные (quote, cond, lambda, eval, apply, funcall).

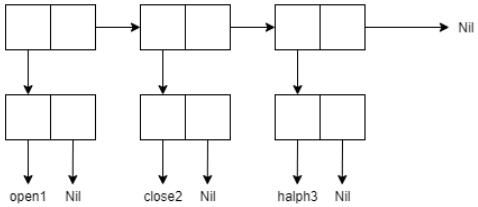
3 Практические задания

3.1 Задание 1

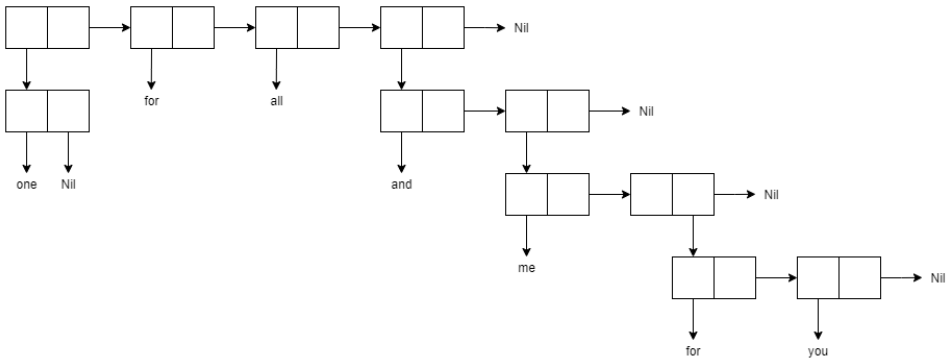
'(open close halph)



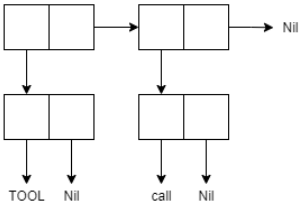
'((open1) (close2) (halph3))



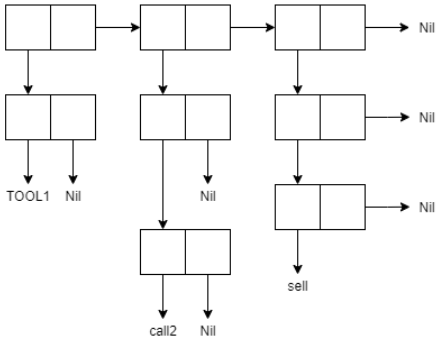
'((one) for all (and (me (for you))))



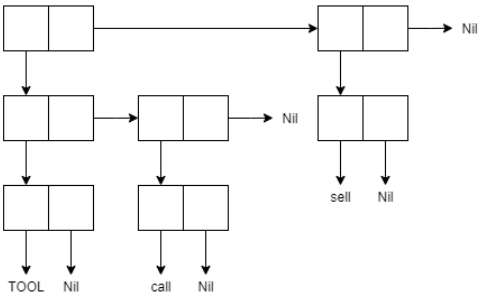
'((TOOL) (calll))



’((TOOL) ((call2)) ((sell)))



’(((TOOL) (call)) ((sell)))



3.2 Задание 2

Обозначим список как <список>. Выражения, возвращающие:

1) второй элемент списка — (car (cdr ’<список>));
2) третий элемент списка — (car (cdr (cdr’ <список>)));
3) четвёртый элемент списка — (car (cdr (cdr (cdr’ <список>)))).

3.3 Задание 3

Определить результат вычисления выражения.

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).

3.4 Задание 4

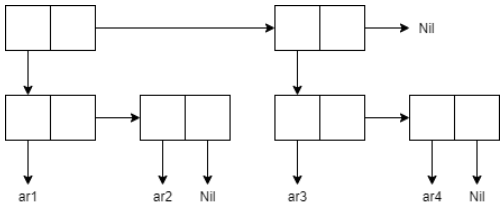
Определить результат вычисления выражения.

1) (list ’Fred ’and ’Wilma) — (Fred and Wilma);
2) (list ’Fred ’(and Wilma) — (Fred (and Wilma));

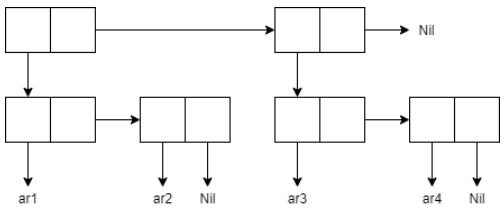
3) (cons Nil Nil) — (Nil);
4) (cons T Nil) — (T);
5) (cons Nil T) — (Nil. T);
6) (list Nil) — (Nil);
7) (cons '(T) Nil) — ((T));
8) (list '(one two) '(free temp)) — ((one two) (free temp));
9) (cons 'Fred '(and Wilma)) — (Fred and Wilma);
10) (cons 'Fred '(Wilma)) — (Fred Wilma);
11) (list Nil Nil) — (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).

3.5 Задание 5

Написать функцию (f ar1 ar2 ar3 ar4), возвращающую список: ((ar1 ar2) (ar3 ar4)).
Лямбда-выражение — (lambda (ar1 ar2 ar3 ar4) (list '(ar1 ar2) '(ar3 ar4))).
Фукнция — (defun f (ar1 ar2 ar3 ar4) (list '(ar1 ar2) '(ar3 ar4)).



Написать функцию (f ar1 ar2), возвращающую ((ar1)(ar2)).
Лямбда-выражение — (lambda (ar1 ar2) (list '(ar1) '(ar2))).
Фукнция — (defun f (ar1 ar2) (list '(ar1) '(ar2))).



Написать функцию (f ar1), возвращающую (((ar1))).
Лямбда-выражение — (lambda (ar1) (list (list '(ar1)))).
Фукнция — (defun f (ar1) (list (list '(ar1)))).