



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

к лабораторной работе №1

*По курсу: «Функциональное и логическое
программирование»*

Тема: «Списки в Lisp».

Студент: Якуба Д.В.

Группа: ИУ7-65Б

Преподаватели: Толпинская Н. Б.,

Строганов Ю. В.

Москва, 2021 г.

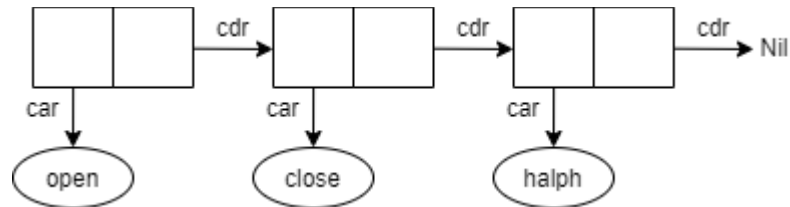
Практическая часть

Задание 1. Представить следующие списки в виде списочных ячеек:

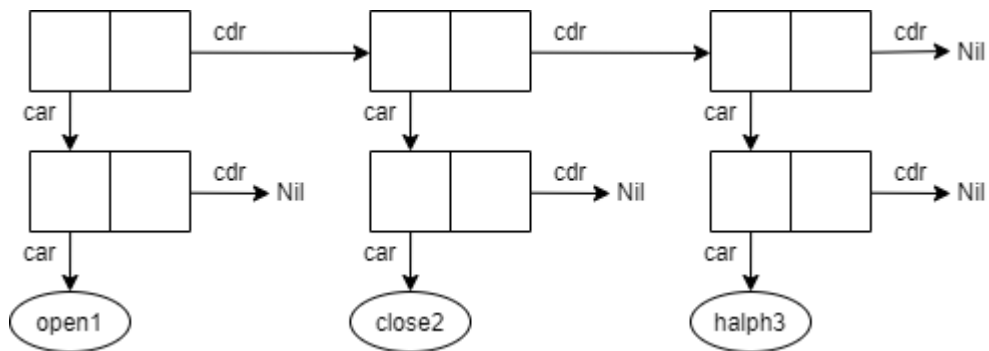
``(open close halph)`
``((open1) (close2) (halph3))`
``((one) for all (and(me(for you))))`

``((TOOL) (call))`
``((TOOL1) ((call2)) ((sell)))`
``(((TOOL) (call)) ((sell)))`

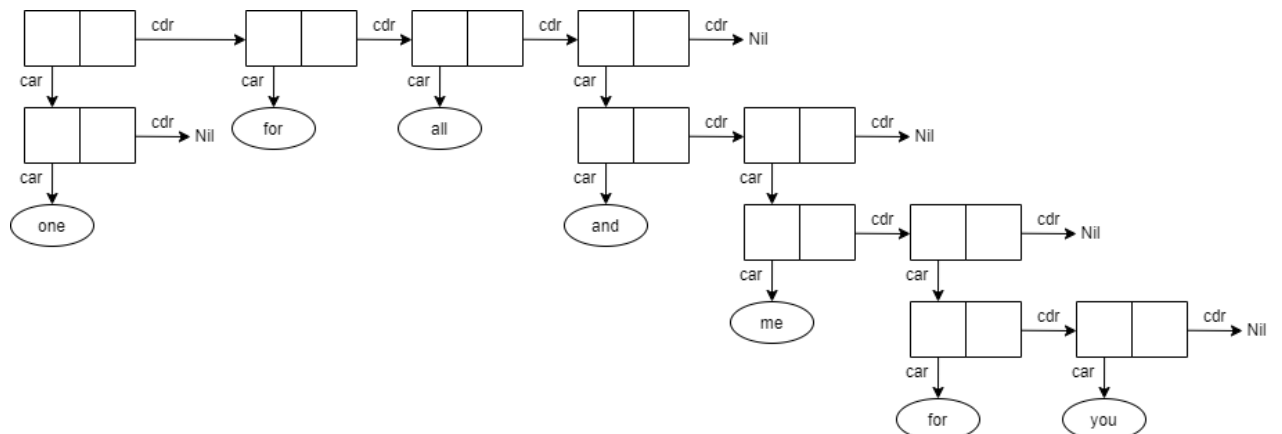
1. ``(open close halph)`



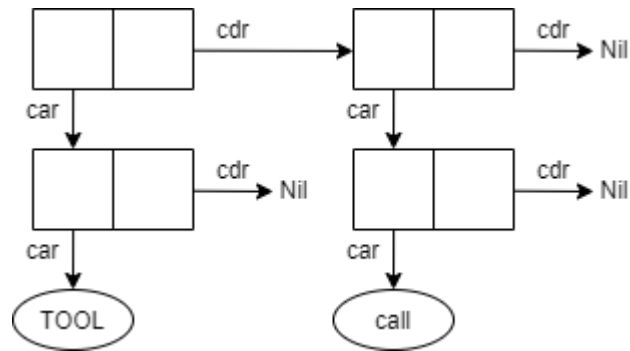
2. ``((open1) (close2) (halph3))`



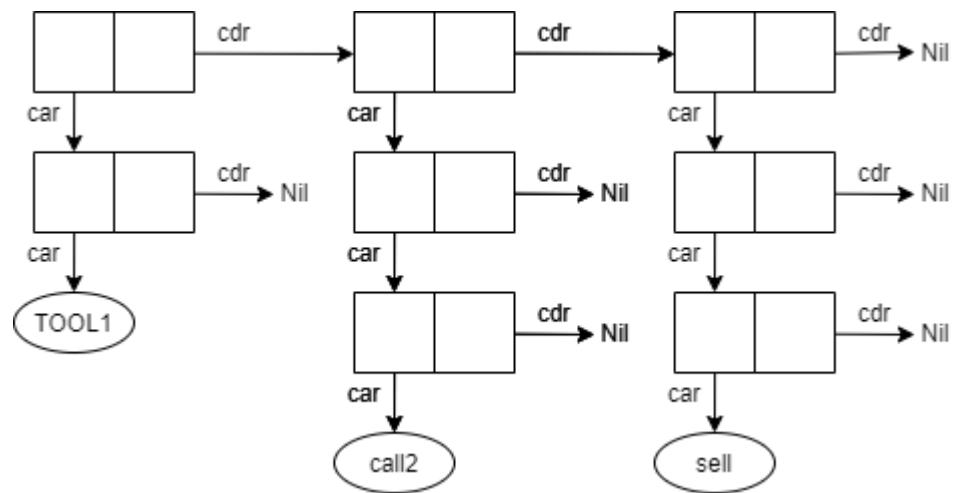
3. ``((one) for all (and(me(for you))))`



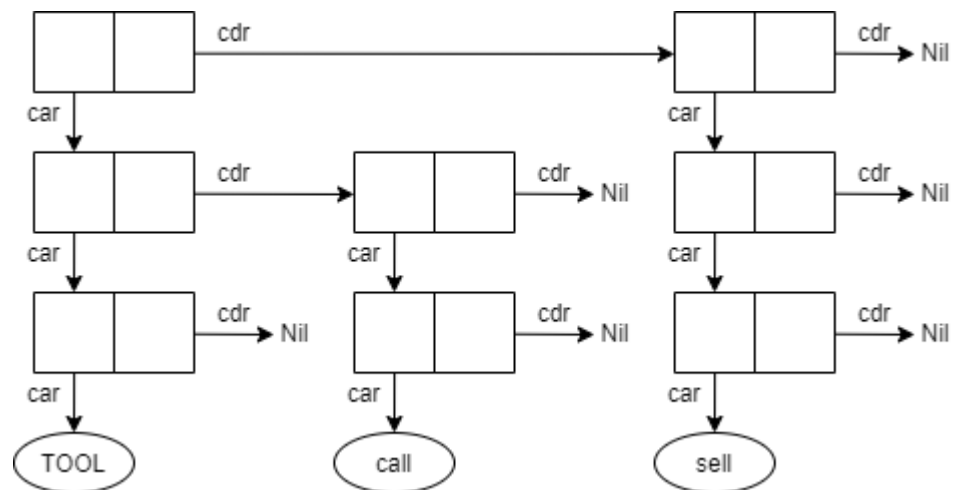
4. `((TOOL) (call))



5. `((TOOL1) ((call2)) ((sell)))



6. `(((TOOL) (call)) ((sell)))



Теоретическая часть

1. Элементы языка Lisp. Перечислить эти элементы.

Элементы языка Lisp включают в себя:

- Атомы:
 - символы (идентификаторы) – синтаксически – набор литер (букв латинского алфавита и цифр), начинающихся с буквы;
 - специальные символы – {T, Nil} (используются для обозначения «логических» констант);
 - самоопределимые атомы – натуральные числа, дробные числа, вещественные числа, строки – последовательность символов, заключенных в двойные апострофы (например “abc”);
- Более сложные «данные» - точечные пары (структуры), которые строятся с помощью унифицированных структур – блоков памяти – бинарных узлов (то есть бинарный узел соответствует минимально адресуемому пространству памяти).

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

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

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

<пустой список> ::= () | Nil,

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

<первый элемент> ::= <S-выражение>,

<хвост> ::= <список>.

Список – это частный случай S-выражения.

2. Синтаксис элементов языка и их представление в памяти.

(мы тут говорили про то, что атом – это пять указателей...)

Любая структура (точечная пара или список) заключается в круглые скобки:

(A . B) – точечная пара, (A) – список из одного 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 C)). Таким образом, синтаксически наличие скобок является признаком структуры – списка или точечной пары.

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

Точечная пара в памяти представляется бинарным узлом.

Атом в памяти представлен пятью указателями.

3. Как воспринимается символьный апостроф?

В зависимости от контекста одни и те же объекты могут играть роль переменных или констант, причем значения и того, и другого могут быть произвольной сложности. Если объект играет роль константы, то для объявления константы достаточно заблокировать его вычисление, то есть как бы взять его в кавычки, отмечающие буквально используемые фразы, не требующие обработки. Для такой блокировки вводится специальная функция `quote`, предохраняющая свой единственный аргумент от вычисления. Апостроф – сокращённое обозначение функции `quote`.

4. Что такое рекурсия? Где используется рекурсия в Lisp?

Рекурсия — это ссылка на определяемый объект во время его определения. Т. к. в Lisp используются рекурсивно определенные структуры (списки), то рекурсия — это естественный принцип обработки таких структур.

Рекурсия используется для организации повторяющихся вычислений. На ней же основано разбиение проблемы и разделение её на подзадачи, решение которых, насколько это возможно, пытаются свести к уже решённым или в соответствии с основной идеей рекурсии к решаемой в настоящий момент задаче. Таким образом, рекурсия в Lisp – организация вычислений и методология (особые отличительные принципы) решения задач.