Министерство образования Российской Федерации

Московский Государственный Технический Университет им. Н.Э. Баумана

Отчет по лабораторной работе №2

По курсу «Функциональное и логическое

программирование»

Студент Медведев А.В.

Группа ИУ7-62 Преподаватель Толпинская Н.Б.

# Теоретическая часть

## Базис Lisp.

* Базис - атомы и структуры (бинарные узлы), хранящие два указателя.
* Базис включает базисные функции и базисные функционалы (atom, eq, car, cdr, cons, quote, cond,lambda, lable,eval)
* Над базисом строятся простые формулы в виде списков где первый аргумент является именем функции, а остальные аргументы ее аргументы.
* Базис это совокупность языка, без которого нет языка (минимальный набор средств языка)

## Классификация функций в Lisp.

* 1. Чистые или строго-математические (Имеют фиксированное кол-во аргументов и один результат)
  2. Специальные (формы) (Могут иметь произвольное количество аргументов; Не все аргументы обрабатываются одинаково)
  3. Псевдо-функции (Функции обеспечивающие некоторый эффект аппаратуры)
  4. Функции допускающие вариационные значения (Позволяют реализовывать логические преобразования)
  5. Функции, допускающие ленивые вычисления (Вызов функций сводится к представлению правила выполнения в зависимости от контекста)
  6. Функции высших порядков (функционалы) (Используются для синтаксического конструирования программ; В качестве аргументов используются функциональные объекты)

## Как представляются списки в ОП?

Список - это особый вид S-выражения, который может быть пустым/не пустым, если он не пустой, то он имеет первый элемент (голову) и хвост (является списком). S-выражения представлены в виде точечных пар, которые состоят из унифицированных структур - блоков памяти - бинарных узлов. Каждый бинарный узел имеет небольшой объем, достаточный для хранения двух типизированных указателей (CAR и CDR, левый и правый, голова и хвост). Пара из первого элемента списка ("голова") и остальных элементов списка ("хвост") представляют собой пару указателей на атомы

## Как выполняются функции CAR и CDR, какие результаты они вернут в разных случаях?

Базовыми функциями доступа к данным в lisp являются функции CAR и CDR.

* CAR - В качестве аргумента принимает список (1 аргумент) и возвращает первый элемент списка, если список не пустой и Nil - иначе
* CDR - в качестве аргумента принимает список (1 аргумент) и возвращает список, состоящий из всех элементов кроме первого, если список не пустой и Nil – иначе

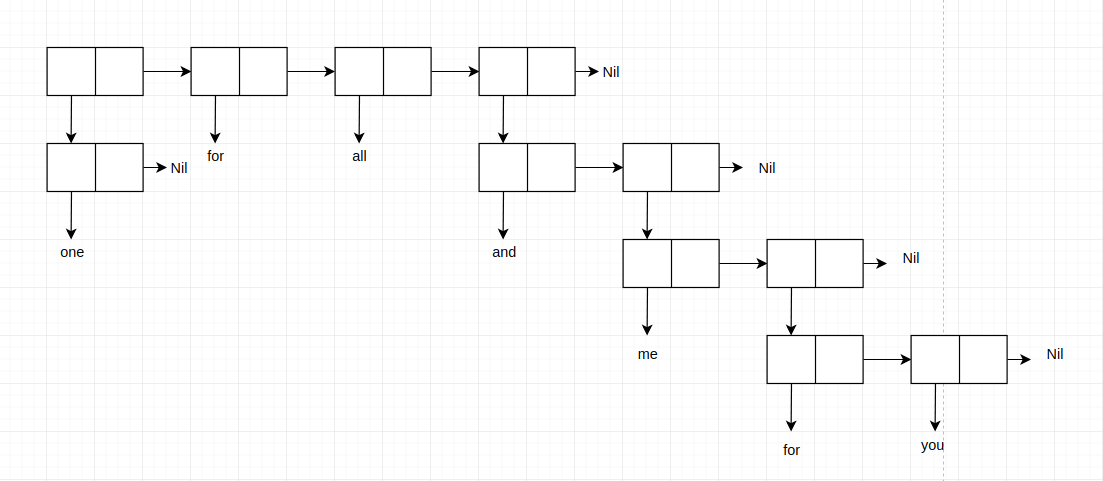
# Практическая часть

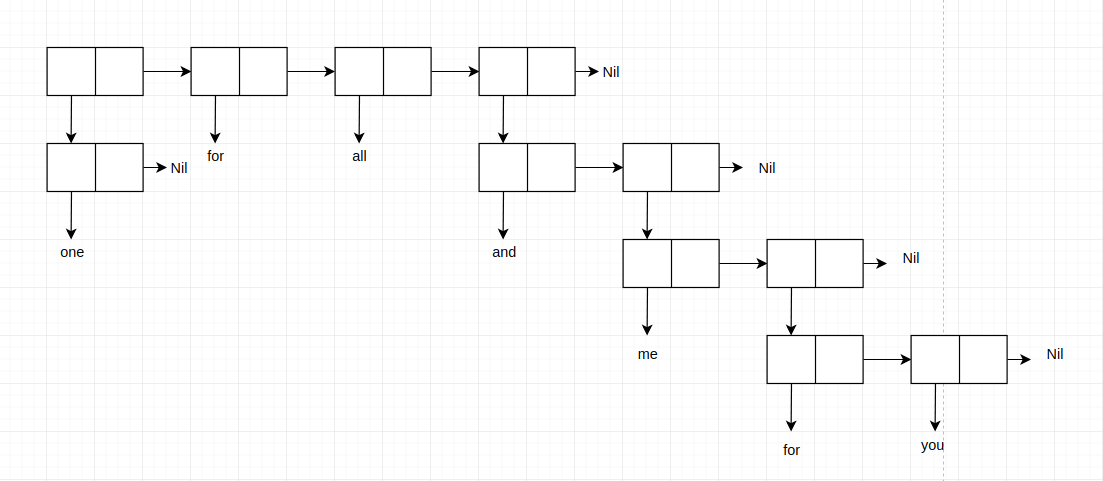
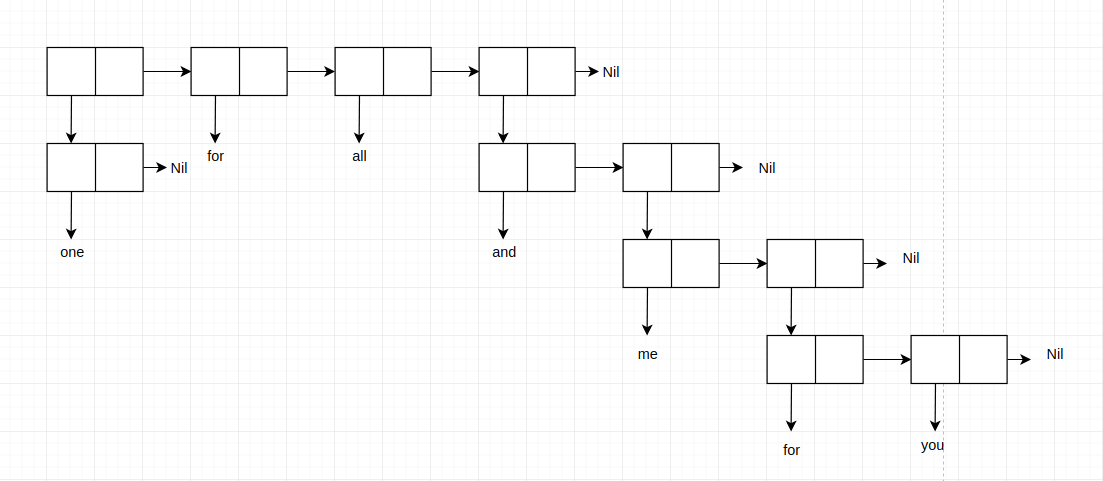
Используя только функции car и cdr,написать выражения, возвращающие:

* 1. второй
  2. третий
  3. четвертый элементы заданного списка.

|  |  |  |
| --- | --- | --- |
|  | Выражение | Результат |
| Второй элемент списка | (CAR (CDR '(1 2 3 4 5))) | 2 |
| Третий элемент списка | (CAR (CDR (CDR '(1 2 3 4 5)))) | 3 |
| Четвертый элемент списка | (CAR (CDR (CDR (CDR '(1 2 3 4 5))))) | 4 |

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

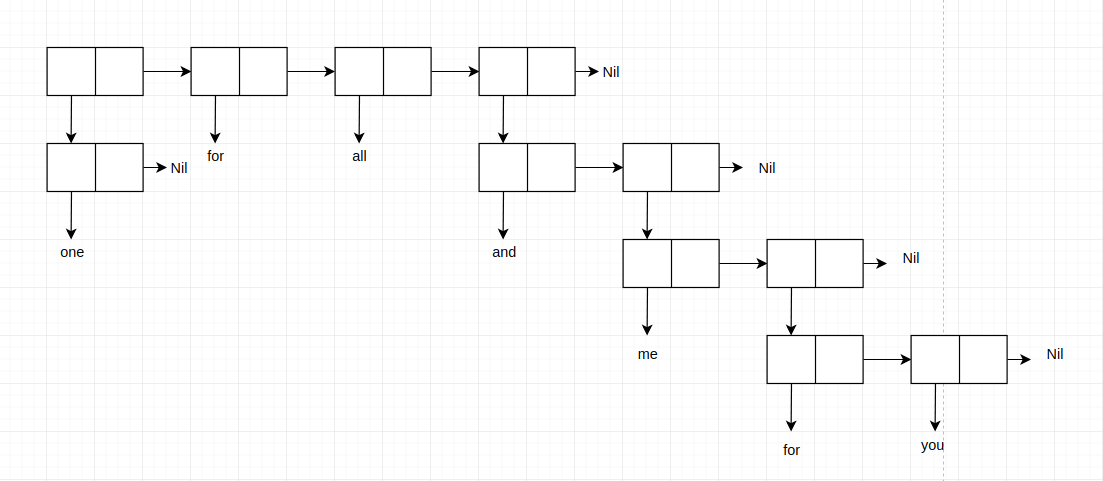
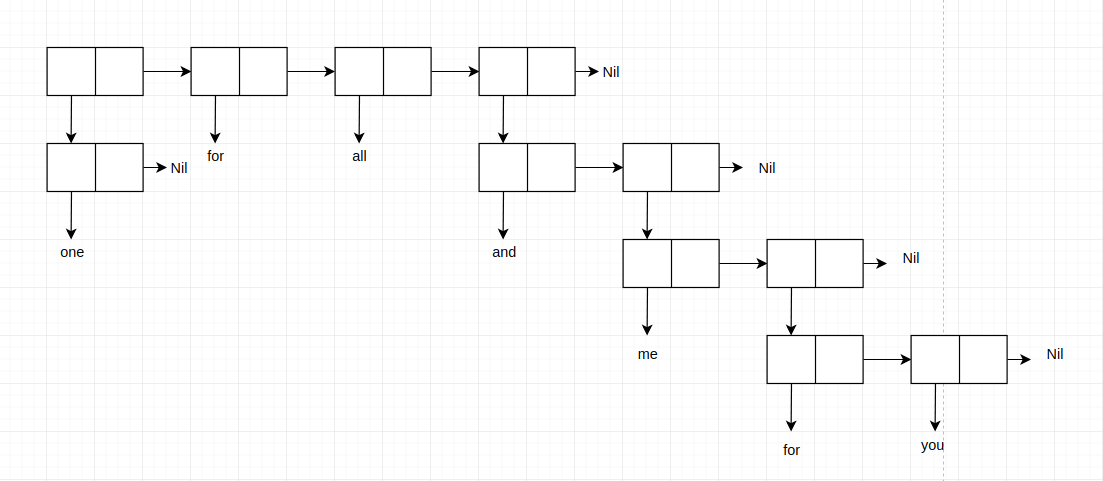




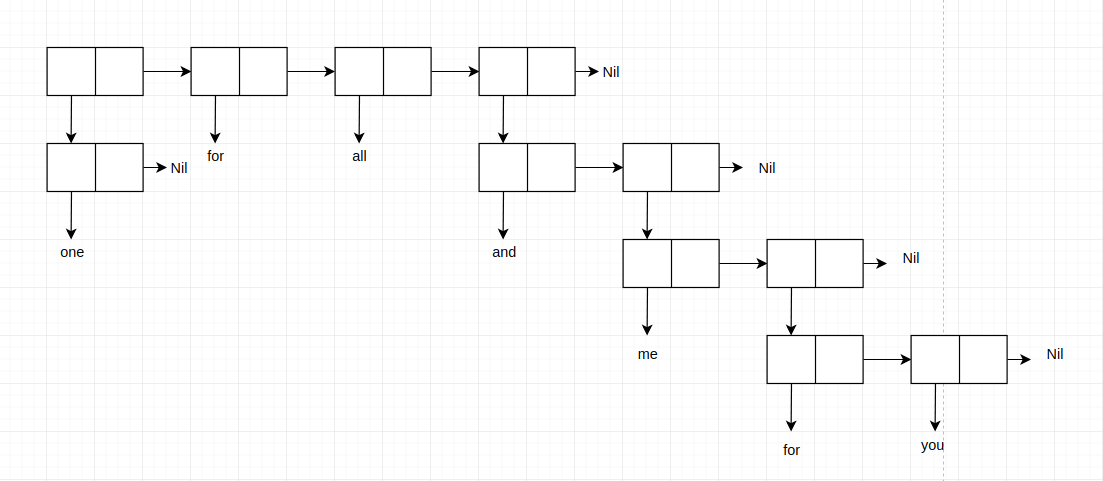
=>

=> for

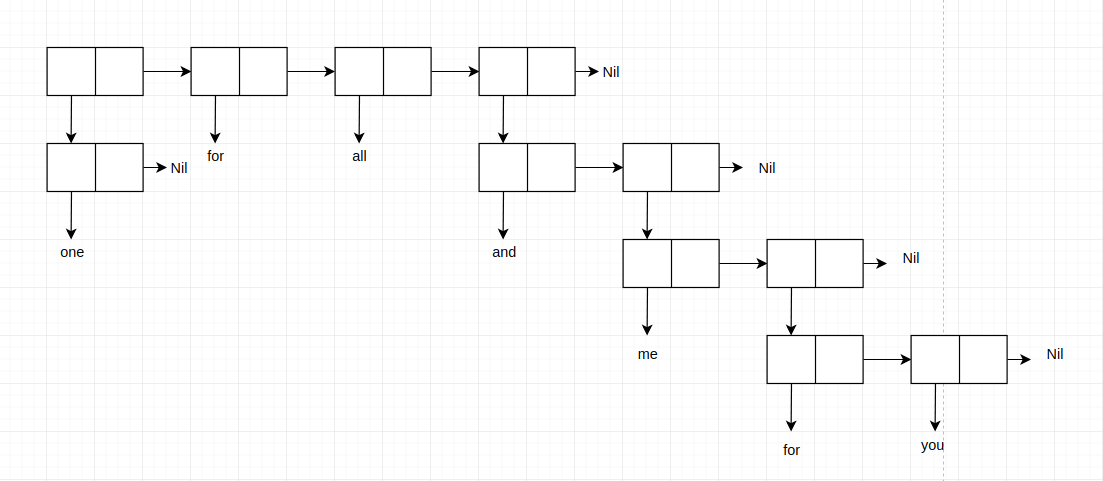
CADDDR '((one) for all (and(me(for you)))) = (AND (ME (FOR YOU)))



=>

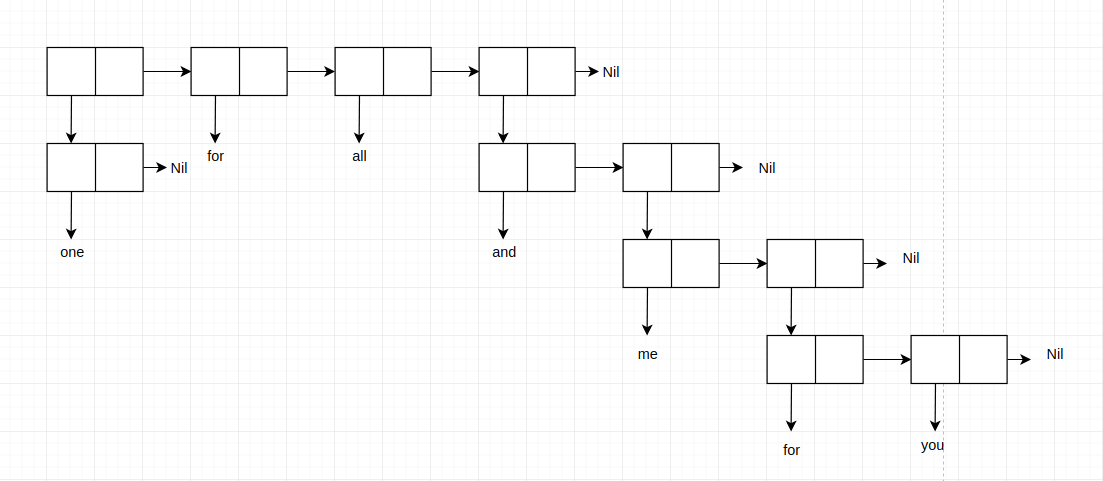
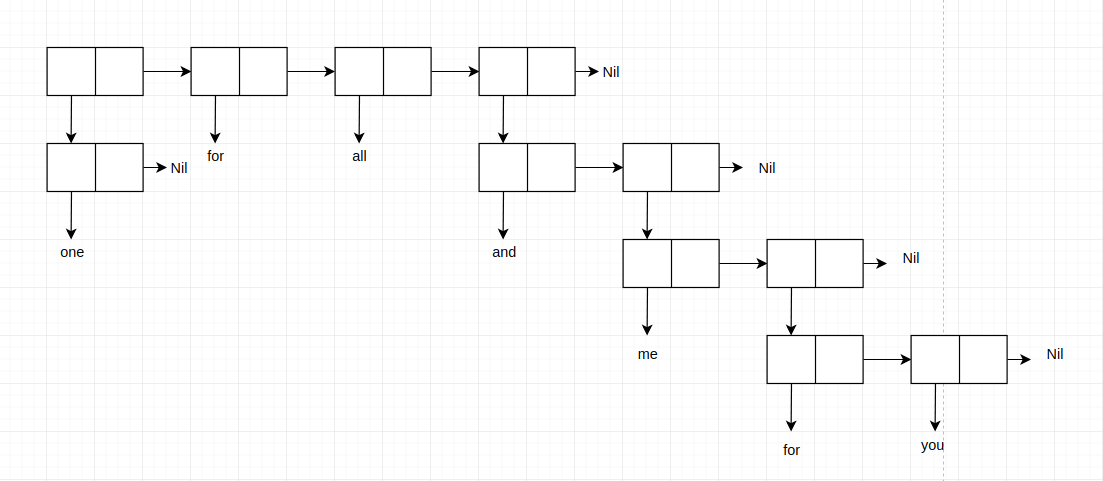


=>

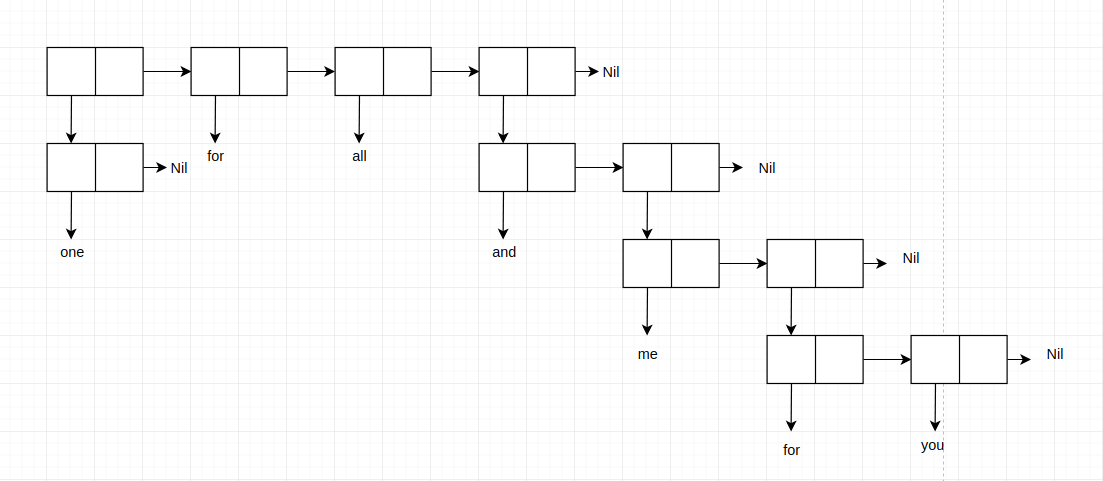


=> (AND (ME (FOR YOU)))

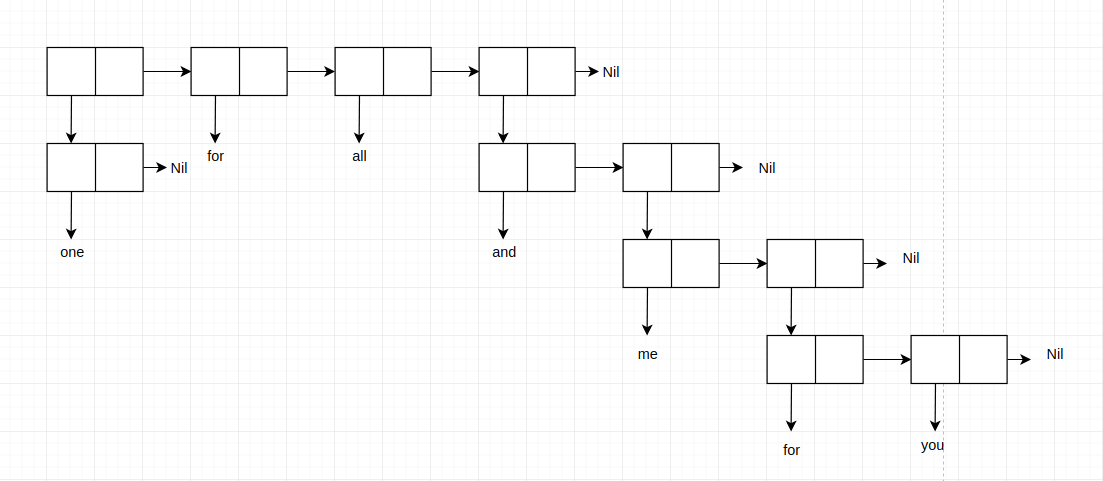
sixth '((one) for all (and(me(for you)))) = NIL



=>



=>



=> Nil

Что будет в результате вычисления выражений

|  |  |  |  |
| --- | --- | --- | --- |
| **Выражение** |  |  | **Результат** |
| (CAADR'((blue cube) (red pyramid))) | ((red pyramid)) | (red pyramid) | **red** |
| (CDAR '((abc) (def) (ghi))) | (abc) |  | **Nil** |
| (CADR '((abc) (def) (ghi))) | ((def) (ghi)) | ((def) | **(def)** |
| (CADDR'((abc)(def)(ghi))) | ((def) (ghi)) | ((ghi)) | **(ghi)** |