



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ _____ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

Название: _____ Использование управляющих структур, работа со списками

Дисциплина: _____ Функциональное и логическое программирование

Студент	<u>ИУ7-64Б</u>	_____	<u>С. Д. Параскун</u>
	Группа	Подпись, дата	И. О. Фамилия

Преподаватель	_____	<u>Н. Б. Толпинская</u>
	Подпись, дата	И. О. Фамилия

Преподаватель	_____	<u>Ю. В. Строганов</u>
	Подпись, дата	И. О. Фамилия

Москва, 2022 г.

1. Теоретические вопросы

1.1 Синтаксическая форма и хранение программы В ПАМЯТИ

В Lisp программа синтаксически представлена в форме S-выражений. Особенностью является единая форма фиксации (отсутствие разделения на программу и данные). И то, и другое представляется списочной структурой, имеющей одинаковую форму. Благодаря такому подходу возможно изменение кода программы при обработке данных.

Так как программа имеет вид S-выражения, в памяти она представлена либо как атом (5 указателей, которыми представляется атом в памяти), либо как списковая ячейка (2 указателя, бинарный узел).

1.2 Трактовка элементов списка

При обработке списков первый элемент воспринимается интерпретатором как название функции, все остальные – ее аргументы. Количество элементов, не считая первого – названия функции, должно совпадать с количеством входных аргументов указанной функции.

В случае если перед скобкой стоит блокировка (‘ или ’), вычисления не производятся, и результатом является все, что стоит после блокировки.

```
1 (defun mult (a b c) (* a b c))
2 (mult 1 2 3) ; => 6; mult - name of func; 1, 2, 3 - arguments
3 (mult 0 1 2 3) ; => error - invalid number of arguments
4 '(eval func 3) ; => (eval func 3)
```

1.3 Порядок реализации программы

1. Ожидает ввода S-выражения.
2. Передает введенное S-выражение функции eval.
3. Выводит полученный результат.

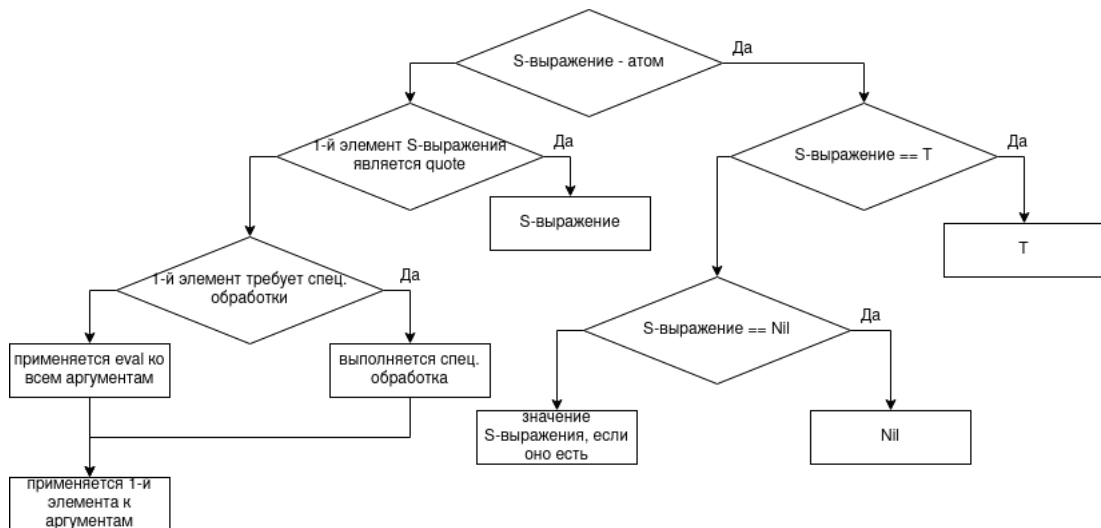


Рисунок 1.1 – Диаграмма работы функции eval

1.4 Способы определения функции

1. С помощью lambda. После ключевого слова указывается лямбда-список и тело функции.

```
1 (lambda (x y) (+ x y))
```

Для применения используются лямбда-выражения.

```
1 ((lambda (x y) (+ x y)) 1 2)
```

2. С помощью defun. Используется для неоднократного применения функции (в том числе рекурсивного вызова).

```
1 (defun sum (x y) (+ x y))
2 (sum 1 2)
```