



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 по лабораторной работе №1 по курсу «Функциональное и логическое программирование»

Тема Списки в Lisre. Использование стандарта.

Студент Козлова И.В.

Группа ИУ7-62Б

Оценка (баллы) _____

Преподаватель Толплинская Н.Ю.

Преподаватель Строганов Ю.В.

Москва — 2022 г.

Ответы на вопросы

1. Элементы языка.

Элементами языка являются атомы и точечные пары.

Атомы представляю из себя:

1. Символы - синтаксически представляется как набор букв и цифр, начинающийся с буквы.
2. Специальные символы - {T, Nil}.
3. Самоопределимые атомы - натуральные, дробные и вещественные числа, а также строки, заключенные в двойные апострофы.

Атомы обычно выглядит как последовательность букв или цифр.

Примеры атомов:

1	B
2	CAT
3	123
4	2/3
5	"abc"

Специальные символы:

1. **T** - Константа. обозначает логическое значение "истина". Истинным значением является все, что отличное от Nil.
2. **Nil** - "ложь". Также обозначает пустой список. Записи nil и () эквивалентны. Являются синтаксисом пустого списка

Точечная пара - (A . B). Строится с помощью бинарных узлов.

1	Точечная пара ::= (<атом>.<атом>)
2	(<атом>.<точечная пара>)
3	(<точечная пара>.<атом>)
4	(<точечная пара>.<точечная пара>)

Пример точечной пары:

1 (A . (B . (C . (D . Nil))))

Облегченная форма записи:

1 (A B C D)

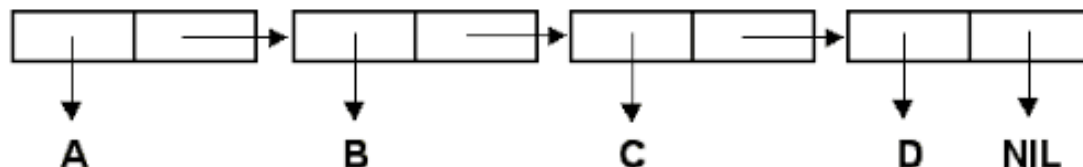


Рисунок 1 – Представление в памяти (A B C D).

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

Список является частым случаем S-выражения.

Список - динамическая структура данных, которая может быть пустой или непустой. Если она не пустая, то состоит из двух элементов:

1. Головы – S-выражение.
2. Хвоста – список.

Список представляет из себя заключенную в скобки последовательность из атомов, разделенных пробелами, или списков. Любой список является программой - его нужно вычислять.

Структуры – точечная пара и список

2. Синтаксис.

Lisp является регистронезависимым языком. Lisp использует префиксную нотацию.

Универсальным разделителем, между атомами, является пробел. В начальных версиях была предложена запятая, но она не прижилась.

Наличие скобок является признаком структуры – списка или точечной пары.

Любая структура заключается в круглые скобки.

(A . B) – точечная пара.

(A) – список из одного элемента.

() или Nil – пустой список.

Одноуровневый список:

1

```
(A B C D)
```

Структурированный список:

1

```
(A (B C) (D E))
```

3. Особенности языка Lisp. Символ апостроф

От других языков программирования Лисп отличается следующими свойствами:

1. одинаковая форма данных и программ;
2. хранение данных, не зависящее от места;
3. автоматическое и динамическое управление памятью;
4. функциональная направленность;
5. Лисп является бестиповым языком;
6. интерпретирующий и компилирующий режимы работы;
7. пошаговое программирование;
8. единый системный и прикладной язык программирования.

Программы, написанные на Лисп, представляются в виде списков.

Символ апостроф – синоним `quote`.

quote – блокирует вычисление своего аргумента. В качестве своего значения выдаёт сам аргумент, не вычисляя его. Перед константами – числами и атомами `T`, `Nil` можно не ставить апостроф.

Пример использования `quote`:

1

```
(quote (car (A B C))) => (car (A B C))
```

Вычисление начинается с внешней функции `quote`, которая возвращает аргумент в неизменном виде.

4. Базис Lisp

Базис Lisp

- атомы и структуры (представляющиеся бинарными узлами);
- базовые (несколько) функций и функционалов: встроенные — примитивные функции (`atom`, `eq`, `cons`, `car`, `cdr`); специальные функции и функционалы (`quote`, `cond`, `lambda`, `eval`, `apply`, `funcall`).

- `atom` проверяет, является ли объект, переданный в качестве аргумента, атомом.

```
1      (atom 'a) ;; t
2      (atom '(a b c)) ;; nil
```

- `eq` проверяет идентичность двух символов

```
1      (eq 'a 'b) ;; nil
2      (eq 'a 'a) ;; t
```

- `cond` (от англ. condition – условие). Нет фиксированного количества аргументов.

Каждый аргумент это список, голова которого рассматривается как условие, в случае если оно истинно, то хвост рассматривается как результат.

```
1      (cond ((eq 'A 'B) 'yes)
2            (T 'not)) ;; NOT
```

- `eval` выполняет двойное вычисление своего аргумента.

```
1      (eval (cons (quote car) (quote ('(A B)))))
```

- `apply` функция, аргументом которой является функция. Данная функция требует функцию и список ее аргументов и возвращает результат этой функции с заданными аргументами.

```
1      (apply #'+ '(1 2 3))
2      (apply #'+ 1 2 '(1 2 3))
```

- `funcall` аналогично `apply`, но не требует, чтобы аргументы были упакованы в список.

1	<code>(funcall #' + 1 2 3)</code>
2	<code>(funcall #' + 1 2 1 2 3)</code>

Ядро – основные действия, которые наиболее часто используются. Ядро шире, чем базис.

5. Что такое рекурсия и примеры из lisp

Рекурсия - это ссылка на описываемый объект в процессе его описания.

Примером рекурсии в lisp служит S-выражение, которое может быть атомом, либо заключенная в скобки пара состоящая из S-выражений (разделенных точкой).

Примером может служить еще список, так как непустой список состоит из головы и хвоста, а хвост в свою очередь может быть списком.

6. Функция

Функцией называется правило, по которому каждому значению одного или нескольких аргументов ставится в соответствие конкретное значение результата.

Функция в Лиспе есть однозначное отображение множества исходных данных на множество её значений. У функции может быть произвольно много аргументов, от нуля до любого конечного числа, но обязательно должно быть хотя бы одно значение

7. Классификация функций

Классификация базисных функций:

- селекторы (`car` `cdr`)

- конструкторы (`cons list`)
- предикаты (`atom`, `consp`, `listp`, `null`, `numberp`, `oddp`)

8. CAR и CDR

Список в языке Lisp представлен одним бинарным узлом, который хранит два указателя (на голову и хвост).

CAR и CDR являются базовыми функциями доступа к данным.

CAR принимает точечную пару или пустой список в качестве аргумента и возвращает первый элемент или `nil`, соответственно.

CDR принимает точечную пару или пустой список и возвращает список состоящий из всех элементов, кроме первого. Если в списке меньше двух элементов, то возвращается `Nil`.

9. LIST и CONS

LIST и CONS являются функциями создания списков (`cons` – базовая, `list` – нет). Функция `cons` (принимает только два аргумента) создает списочную ячейку и устанавливает два указателя на аргументы.

Функция `list` принимает переменное число аргументов и возвращает список, элементы которого – переданные в функцию аргументы.

Например список `'(open close halph)` из задания 1 можно представить как: `(cons 'open (cons 'close (cons 'halph nil)))` или `(list 'open 'close 'halph)`.