

Московский государственный технический университет
имени Н. Э. Баумана



Факультет: Информатика и системы управления

Кафедра: Программное обеспечение ЭВМ и информационные технологии

Дисциплина: Функциональное и логическое программирование

Лабораторная работа №4

Выполнили: Никичкин А. С., Фокеев А. И.
Группа: ИУ7–61

Москва, 2015 г.

1 Чем принципиально отличаются функции CONS, LIST, APPEND

Пусть

```
(setf lst1 '(a b))  
(setf lst2 '(c d))
```

Тогда при использовании CONS

```
(cons lst1 lst2) => ((A B) C D)
```

Таким образом CONS создаёт *списочную ячейку*, у которой *car-указатель* указывает на первый аргумент, а *cdr-указатель* — на второй. При использовании LIST

```
(list lst1 lst2) => ((A B) (C D))
```

Получаем, что LIST гарантирует возвращение нового *списка* из элементов, переданных функции в качестве аргументов. При использовании APPEND

```
(append lst1 lst2) => (A B C D)
```

APPEND возвращает новый список, который формируется из объединения списков переданных функции в качестве аргументов. При этом последний аргумент может являться не списком. CDR указатель нового списка будет указывать как раз на последний аргумент.

2 Вычислить результат выражения

Задание 2.1

```
(reverse ()) => NIL
```

Задание 2.2

```
(last ()) => NIL
```

Задание 2.3

```
(reverse '(a)) => (A)
```

Задание 2.4

```
(last '(a)) => A
```

Задание 2.5

```
(reverse '((a b c))) => ((A B C))
```

Задание 2.6

```
(last '((A B C))) => ((A B C))
```