

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 №1

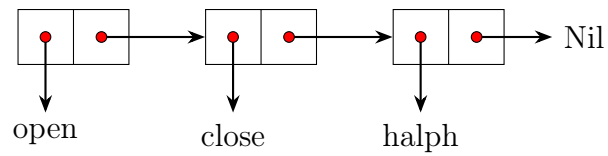
Выполнили: Никичкин А. С., Фокеев А. И.

Группа: ИУ7–61

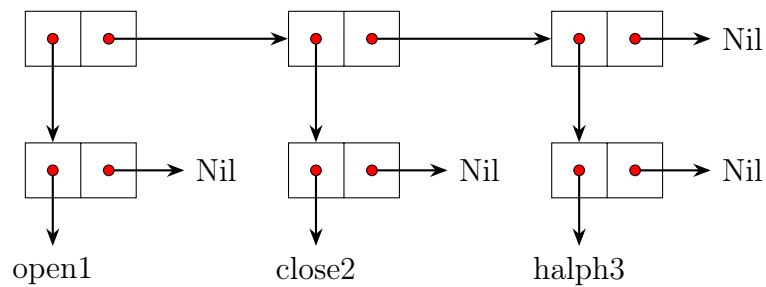
Москва, 2015 г.

1 Представить списки в виде списочных ячеек

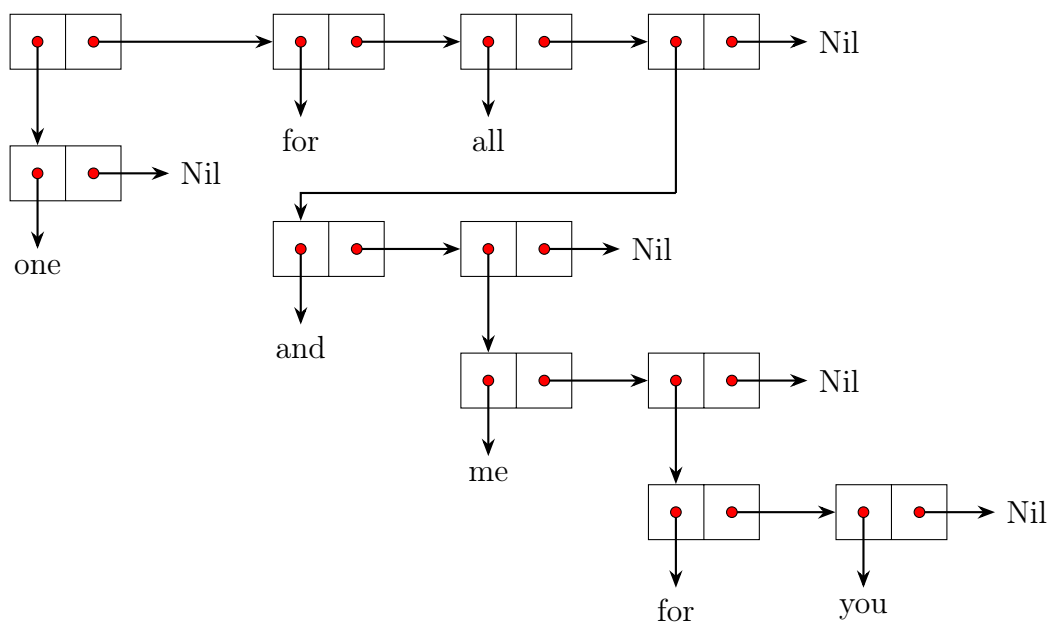
Задание 1.1 $'(open\ close\ halph)$



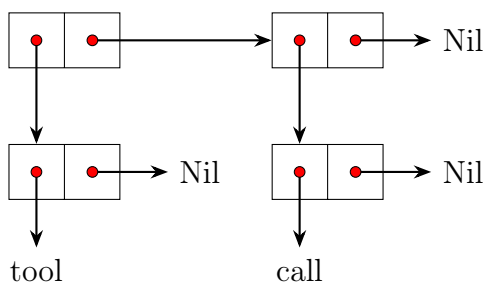
Задание 1.2 $'((open1)\ (close2)\ (halph3))$



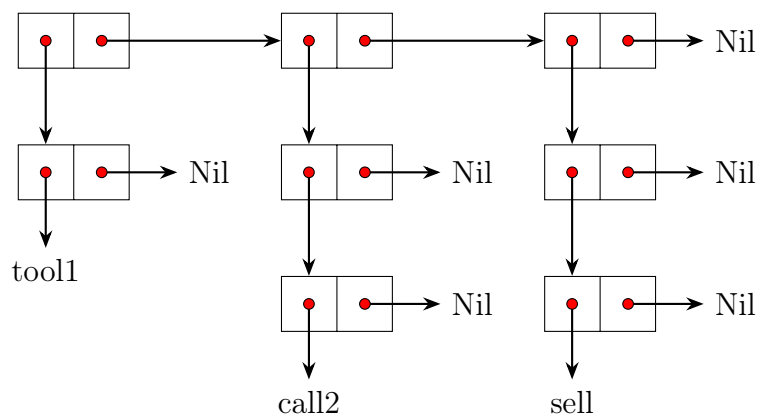
Задание 1.3 $'\left((one)\ for\ all\ \left(and\ (me\ (for\ you))\right)\right)$



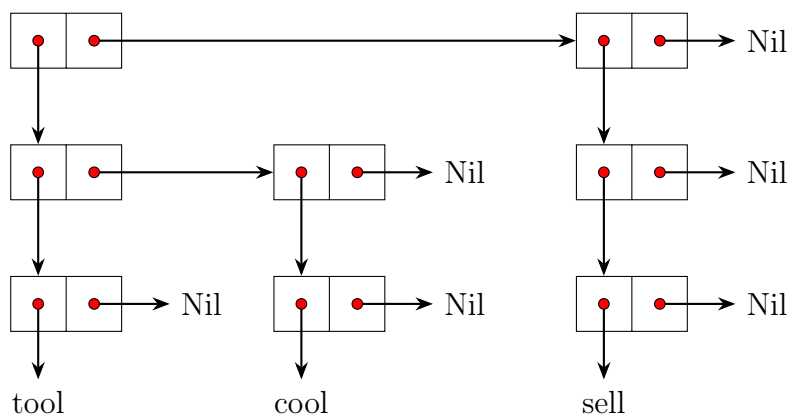
Задание 1.4 $'((tool) (call))$



Задание 1.5 $'((tool1) ((call2)) ((sell)))$



Задание 1.6 $'(((tool) (call)) ((sell)))$



2 Используя только функции CAR и CDR, написать выражение

Задание 2.1 Возвращающие второй элемент списка

```
(car (cdr example-list)) = (cadr example-list)
```

Задание 2.2 Возвращающие третий элемент списка

```
(car (cdr (cdr example-list))) = (caddr example-list)
```

Задание 2.3 Возвращающие четвёртый элемент списка

```
(caddr example-list)
```

3 Вычислить результат выражения

Задание 3.1

```
(caadr '((blue cube) (red pyramid))) => RED
```

Задание 3.2

```
(cdar '((abc) (def) (ghi))) => NIL
```

Задание 3.3

```
(cadr '((abc) (def) (ghi))) => (DEF)
```

Задание 3.4

```
(caddr '((abc) (def) (ghi))) => (GHI)
```

Задание 3.5

```
(list 'Fred 'and Wilma) => Unbound variable WILMA
```

Задание 3.6

```
(list 'Fred '(and Wilma)) => (FRED (AND WILMA))
```

Задание 3.7

```
(cons Nil Nil) => (NIL)
```

Задание 3.8

```
(cons T Nil) => (T)
```

Задание 3.9

```
(cons Nil T) => (NIL . T)
```

Задание 3.10

```
(list Nil) => (NIL)
```

Задание 3.11

```
(cons (T) Nil) => Undefined function T
```

Задание 3.12

```
(list '(one two) '(free temp)) => ((ONE TWO) (FREE TEMP))
```

Задание 3.13

```
(cons 'Fred '(and Wilma)) => (FRED AND WILMA)
```

Задание 3.14

```
(cons 'Fred '(Wilma)) => (FRED WILMA)
```

Задание 3.15

```
(list Nil Nil) => (NIL NIL)
```

Задание 3.16

```
(list T Nil) => (T NIL)
```

Задание 3.17

```
(list Nil T) => (NIL T)
```

Задание 3.18

```
(cons T (list Nil)) => (T NIL)
```

Задание 3.19

```
(list (T) Nil) => Undefined function T
```

Задание 3.20

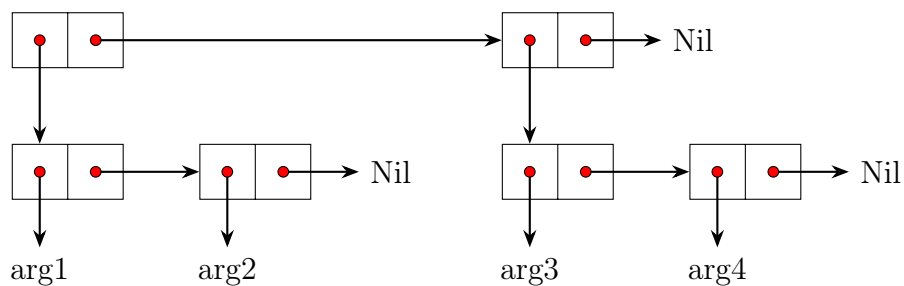
```
(cons '(one two) '(free temp)) => ((ONE TWO) FREE TEMP)
```

4 Написать функции и представить результаты в виде списочных ячеек

Задание 4.1 Функция от четырёх аргументов возвращающая список вида

$$((arg1\ arg2)\ (arg3\ arg4))$$

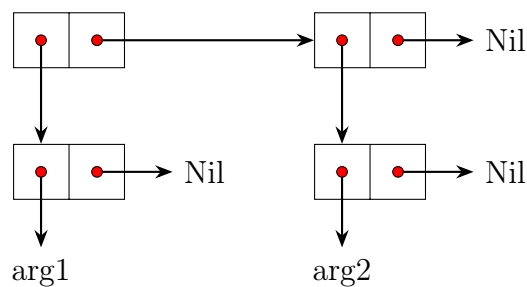
```
1 (defun problem-5-1 (arg1 arg2 arg3 arg4)
2   "Make list like ((arg1 arg2) (arg3 arg4))"
3   (list (list arg1 arg2)
4         (list arg3 arg4)))
```



Задание 4.2 Функция от двух аргументов возвращающая список вида

$$((arg1)\ (arg2))$$

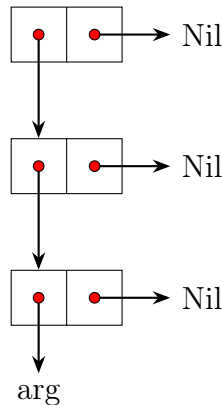
```
1 (defun problem-5-2 (arg1 arg2)
2   "Make list like ((arg1) (arg2))"
3   (list (list arg1)
4         (list arg2)))
```



Задание 4.3 Функция от одного аргумента возвращающая список вида

$$(((arg)))$$

```
1 (defun problem-5-3 (arg)
2   "Make list like (((arg)))"
3   (list (list (list arg))))
```



5 Заключение

В данной лабораторной работе было изучено понятие *списочной ячейки* и составление *диаграммы представления списка в виде списочных ячеек*. Помимо определения новой функции при помощи макроса **defun**, так же были рассмотрены:

- специальный оператор **quote** или **'** — блокирует вычисление выражения;
- функция **car** — возвращает *car-часть списочной ячейки* или если рассматривается список, то говорят о *первом элементе списка*, который является S-выражением т.е. *голова списка*;
- функция **cdr** — возвращает *cdr-часть списочной ячейки* или если рассматривается список, то говорят о *списке из оставшихся элементов* за исключением первого т.е. *хвост списка*;
- функция **cons** — создаёт новую *списочную ячейку*, у которой *car-часть* принимает первый аргумент, а *cdr-часть* — второй аргумент;
- функция **list** — возвращает *список* из элементов переданных ей в качестве аргументов.