

В наследство от функционального программирования, и Lisp в частности, Maxima получила списки как **базовую форму представления данных**. Списком является **группа элементов**, разделённых запятыми и заключённых в квадратные скобки (совсем как в Python). Список может быть **пустым** или содержать любые другие элементы, в том числе **вложенные списки**.

В данном буклете вы найдете основные команды для работы со списками в компьютерной программе **Maxima**.

ОСНОВНЫЕ ВОЗМОЖНОСТИ MAXIMA

Работа со списками

Буклет выполнен
БЕЛЕНКО АНАСТАСИЕЙ ВИТАЛЬЕВНОЙ
РГПУ им. А. И. ГЕРЦЕНА
ИИТиТО, ИВТ, 1 КУРС

b.nastya131101@gmail.com



ОСНОВНЫЕ ВОЗМОЖНОСТИ

MAXIMA

ИСПОЛЬЗУЕМЫЕ ДЛЯ РАБОТЫ
СО СПИСКАМИ



Обращение к списку и его элементам:

- 1) **list1[1]** – обращение к первому элементу в списке list1
- 2) **list1** – обращение ко всему списку list1
- 3) **first(list1)** – вывод первого элемента списка list1
- 4) **last(list1)** – вывод последнего элемента списка list1

Добавление и удаление элементов списка:

- 1) **append(list1, list2, listN)** – «склеивает» списки list1, list2, ..., listN в один список
- 2) **join(list1, list2)** – создает новый список, состоящий из поочередно добавленных элементов списков list1 и list2
- 3) **rest(list1, x)** – создает новый список из списка list1 с удалением первых x (если $x > 0$) или последних x (если $x < 0$) элементов
- 4) **unique(list1)** – создает новый список из списка list1 после удаления повторяющихся элементов
- 5) **delete(x, list1)** – удаление всех элементов со значением x из списка list1

Вычисление количества элементов списка:

- 1) **length(list1)** – возвращает длину списка list1

Создание копий списка:

- 1) **copylist(list1)** – возвращает копию списка list1

Создание списка:

- 1) **makelist(expr, x, x0, xn)** – возвращает список из элементов вида expr, где expr зависит от целочисленного параметра x, пробегающего значения от x0 до xn
- 2) **makelist(expr, x, list1)** – возвращает список из элементов вида expr, где expr зависит от целочисленного параметра x, пробегающего значения от x0 до length(list1)
- 3) **create_list(expr, x1, list1, ..., xn, listn)** – строит список путём вычисления выражения form, зависящего от x1, к каждому элементу списка list1 и т.д.

Изменение порядка

следования элементов списка:

- 1) **cons(x, list1)** – создает новый список, первым элементом которого будет x, а далее элементы списка list1
- 2) **endcons(x, list1)** – создает новый список, первыми элементами которого будут элементы списка list1, и последнего элемента – x
- 3) **reverse(list1)** – меняет порядок элементов в списке list1 на обратный



Поиск минимального и максимального значения в списке:

- 1) **apply(max, list1)** – использует функцию max ко всему списку list1 (находит максимум этого списка)
- 2) **apply(min, list1)** – использует функцию min ко всему списку list1 (находит минимум этого списка)

Суммирование и перемножение списков:

- 1) **sum(expr, x, x0, xn)** – суммирование значений выражений expr, зависящих от индекса x, который изменяется в диапазоне от x0 до xn
- 2) **product(expr, x, x0, xn)** – произведение значений выражений expr, зависящих от индекса x, который изменяется в диапазоне от x0 до xn

Трудности при работе со списками:

Иногда использование списка неоправданно. Его заменяют массивом – чаще всего из-за случая с многократным удалением или добавлением новых элементов. При использовании списка к го элементу можно обратиться, только последовательно пройдя все стоящие до него элементы. В массиве же мы ссылаемся на конкретный элемент, не затрагивая при этом другие.