

Основные возможности Maxima, используемые для работы со списками

`length(список)` - определяет количество элементов (длину).

`length(list)` - 3

`copylist(список)` - создаёт копию.

`append(список1, список2, ...)` - склеивает списки.

`reverse(список)` - меняет порядок следования элементов на обратный.

`reverse(list)` - `[[x,y],a,1]`

`member(элемент,список)` - возвращает true, если элемент входит в список, иначе false.

`member(a,list)` - true

Аналоги Lisp-a:)

`first(список)` - первый элемент ("car").

`first(list)` - 1

`rest(список)` - хвост, т.е. всё кроме первого элемента ("cdr").

`rest(list)` - `[a,[x,y]]`

`last(список)` - последний элемент.

`cons(элемент,список)` - добавить элемент в начало (создаёт новый список).

`cons(t,list)` - `[t,1,a,[x,y]]`

`endcons(элемент,список)` - добавить элемент в конец (создаёт новый список).

"Генераторы" списков.

`makelist(выражение, i, a, b)` - создание списка на основе заданного выражения при изменении переменной i от a до b.

`makelist(a/i, i, 1, 4)` - `[a, a/2, a/3, a/4]`

`makelist(выражение, x, список)` - создание списка на основе заданного выражения из исходного списка.

`makelist(a/x, x, [1,2,3,4])` - `[a, a/2, a/3, a/4]`

`create_list(выражение, x1, список1, x2, список2, ...)` - создание списка на основе заданного выражения из нескольких исходных списков.

Действия над элементами списка.

`product(выражение, i, a, b)` - перемножает выражения при изменении i от a до b.

`product(i-x, i, 0, 3)` - `x(x+1)(x+2)(x+3)`

`sum(выражение, i, a, b)` - суммирует выражения при изменении i от a до b.

`sum(x/i, i, 1, 3)` - `x+x/2+x/3+x/4`

`map(функция, выражение1, выражение2, ...)` - применяет функцию (оператор, операцию) к последовательности выражений.

`map("-", [a,b], [1,-1])` - `[a-b, b-a]`

`apply(функция, список)` - применяет функцию к заданному списку (преобразует список в аргументы функции).

`apply(max, [1,5,20])` - 20