



Math-Net.Ru

All Russian mathematical portal

A. R. Esayan, Automatic loading in Maxima documents of definitions of functions and values of variables,  
*Chebyshevskii Sb.*, 2011, Volume 12, Issue 4, 93–96

<https://www.mathnet.ru/eng/cheb112>

Use of the all-Russian mathematical portal Math-Net.Ru implies that you have read and agreed to these terms of use

<https://www.mathnet.ru/eng/agreement>

Download details:

IP: 18.97.14.81

May 23, 2025, 07:55:14



# ЧЕБЫШЕВСКИЙ СБОРНИК

## Том 12 Выпуск 4 (2011)

---

УДК: 511.37 + 511.36

### АВТОЗАГРУЗКА В ДОКУМЕНТЫ *Maxima* ОПРЕДЕЛЕНИЙ ФУНКЦИЙ И ЗНАЧЕНИЙ ПЕРЕМЕННЫХ

А. Р. ЕСАЯН (г. Тула)

Свободно распространяемая система *Maxima* с интерфейсом *wxMaxima* является универсальным математическим пакетом, применимым к решению обширного класса задач. В России, как и во всем мире, постепенно происходит миграция от проприетарного программного обеспечения на свободное программное обеспечение (СПО). В частности, Министерство образования и науки РФ в 2009 г. утвердило перечень СПО, рекомендуемого для использования в школах. В этом списке находится и *Maxima*.

В статье речь идет о ссылках в *wxt*-документах *Maxima* на ресурсы других файлов и использование этих ресурсов в текущих документах. Возможность создания ресурсных ссылок чрезвычайно полезна. Она позволяет структурировать документы, способствует упрощению их сопровождения, и что немаловажно, содействует отказу от создания плохо обзримых документов больших размеров. На простом иллюстративном примере подробно продемонстрирована и объяснена вся технология автозагрузки определений функций и значений переменных для всех возможных типов функций и переменных. Основным средством для этого служит встроенная функция

$$\text{setup\_autoload}(\text{mac\_name}, g_1, g_2, \dots, g_n). \quad (1)$$

По обращению (1), выполненному из текущего документа  $d$ , в  $d$  виртуально подгружаются определения функций и значения переменных из некоторого *Maxima*-документа  $t$ . Виртуальность в данном случае означает, что в самом документе  $d$  загруженные ресурсы не появляются. Они лишь становятся доступными в нем. Документ  $t$  должен иметь расширение  $mac$ , то есть быть демонстрационным файлом. В (1) имя файла  $t$  или полный путь доступа к нему задается строковым аргументом  $mac\_name$ . Последующие аргументы (1) – это имена функций из  $t$ . Заметим, что достаточно указать в (1) имя лишь одной функции, определенной в  $t$ . После первого же обращения в  $d$  к этой функции, доступными для работы в  $d$  окажутся все функции и переменные  $t$ . Это касается простых переменных, объявленных массивов, динамических массивов, обычных функций, макросов и массивовых функций.

Напомним [1, с. 50-51], как создать из документа *t.wxt* текстовый файл *t.mac*. Сделать это можно так:

- сформировать (или открыть) *Maxima*-документ *t.wxt*;
- сохранить его как файл *t.mac* (Файл/*Save As*/задать тип файла – *Maxima-batch file (\*.mac)*/задать имя файла (и путь к нему)/Сохранить).

Проиллюстрируем автозагрузку функций и переменных на примере. Пусть имеется следующий файл *t.wxt* с определениями функций и переменных (ячейки вывода здесь не показаны):

• %i1	ggg1(x) ::= x^2	/* функция (макрос).	A */;
• %i2	ggg2(x, y) := x+y	/* функция.	B */;
• %i3	ggg2(x, y) := x*y*z	/* функция.	C */;
• %i4	xxx : 777	/* простая переменная.	D */;
• %i5	ma : matrix([1, 2, 3], [4, 5, 6])	/* матрица.	E */;
• %i6	li : [1, 2, 3, 4, 5, 6, 7, 8]\$	/* список.	F */;
• %i7	array(ar, 2, 2, 1)\$ fillarray(ar, li)	/* массив.	G */;
• %i8	mas[x] := x^3	/* массивовая функция.	H */;
• %i9		/* динамический массив tor (хэш-массив)	I */;
• %i10	tor[1, 2, 1] : {7, 8}; tor[0, 0, -1] : h;		

Создадим из файла *t.wxt* текстовый демонстрационный файл *t.mac* так, как это было описано выше. Пусть *t.mac* размещен в папке “d:/0/”. Фактически в *t.mac* из *t.wxt* переносятся только определения объектов без комментариев и содержимого ячеек вывода. При просмотре *t.mac* в текстовом редакторе мы увидим лаконичный текст всех определений:

```

ggg1(x)::=x^2;
ggg2(x, y):=x+y;
ggg3(x,y,z) :=x*y*z;
xxx:777;
ma : matrix([1, 2, 3], [4, 5, 6]);
li : [1, 2, 3, 4, 5, 6, 7, 8]$
array(ar, 2, 2, 1)$ fillarray(ar, li)$
mas[x] := x^3;
tor[1, 2, 1] : {7, 8}; tor[0, 0, -1] : h;

```

Файл *t.mac* является демонстрационным и запуск его на демонстрацию можно организовать функцией `demo("d:/0/t.mac")`, а если изменить расширение *mac* на *dem*, то и еще проще – функцией `demo("d:/0/t")`. Чтобы не писать полный путь до пользовательского файла *t.mac* можно разместить *t.mac* в концевой папке на одном из путей доступа к встроенным демонстрационным файлам системы *Maxima*. А список таких путей является значением переменной `file_search_demo`.

Вернемся к нашему примеру. Файл *t.mac* создан. Создадим новый документ *d.wxm* и проведем в нем следующие вычисления:

•	%i1	mac_name : "d:/0/t.mac"	/* A */\$
	%o2	setup_autoload(mac_name, ggg1);	
	%o3	done	
	%i3	functions; values	/* B */;
•	%o3	[]	
	%o4	[mac_name]	
•	%i5	ggg2(5, 6); ggg1(5); ggg3(x, y, z)	/* C */;
	%o5	ggg2(5, 6)	
	%o6	25	
	%o7	x.y.z	
•	%i8	ggg2(5, 6); ggg3(2, 4, 7)	/* D */;
	%o8	11	
	%o9	56	
•	%i10	fundef(ggg1); fundef(ggg2); fundef(ggg3)	/* E */;
	%o10	ggg1(x) ::= x <sup>2</sup>	
	%o11	ggg2(x, y) := y+x	
	%o12	ggg3(x, y, z) := x.y.z	
•	%i13	values	/* F */;
	%o13	[mac_name, xxx, ma, li]	
•	%i14	[mac_name,xxx, ma, li]	/* G */;
	%o14	[d : /0/t.mac, 777, $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ , [1, 2, 3, 4, 5, 6, 7, 8]]	
•	%i15	[mas[3], ar[0, 0, 1], tor[1, 2, 1], tor[0, 0, 0]]	/* H */;
	%o15	[27, 2, {7, 8}, tor <sub>0,0,0</sub> ]	

Прокомментируем вычисления, проведенные в *d.wxm*.

А. В текущий документ *d.wxm* потенциально подгружаются все средства, определенные в документе *t.mac*. Делается это выполнением функции `setup_autoload`. Реальная загрузка этих средств будет произведена при первом же обращении в *d* к функции `ggg1`.

В. Значение переменной *functions* равно пустому списку [], что соответствует отсутствию определений пользовательских функций в текущем документе, а определения функций из *t.mac* пока еще не действуют. Значение переменной *values* равно списку [mac\_name], что соответствует наличию только одной опре-

деленной простой переменной *mac\_name* в текущем документе, а определения переменных из *t.mac* пока еще не действуют.

С. Обращение `ggg2(5, 6)` приводит к выводу невычисленного значения `ggg2(5, 6)` потому, что в *d.wxt* функция `ggg2` еще не действует. Следующее обращение `ggg1(5)` приводит к подключению всех определений *t.mac* к текущему документу *d* и потому для `ggg1(5)` выводится  $25 (5^2)$ , а для `ggg3(x, y, z) - x \cdot y \cdot z (x*y*z)`.

Д. Повторное обращение `ggg2(5, 6)` приводит теперь к выводу  $11 (5+6)$ , а обращение `ggg3(2, 4, 7) -` к выводу  $56 (2*4*7)$ .

Е. По функциям `fundef` выводятся определения трех функций `ggg1`, `ggg2` и `ggg3`, подгруженные из файла *t.mac*.

Ф. Значением переменной *values* является список переменных, действующих в текущем документе, не считая массивов, динамических массивов (массивовых функций).

Г. По списку `[mac_name, xxx, ma, li]` выводится список текущих значения переменных, являющихся компонентами этого списка.

Н. По списку `[mas[3], ar[0, 0, 1], tor[1, 2, 1], tor[0, 0, 0]]` выводятся:  $27 -$  значение массивовой функции `mas[x]` в точке  $3$ ;  $2 -$  элемент из позиции `[0, 0, 1]` объявленного массива *ar*; множество `{7, 8}` – элемент из позиции `[1, 2, 1]` динамического массива *tor* и, наконец, `tor0,0,0` – неопределенный элемент из позиции `[0, 0, 0]` динамического массива *tor*.

Отметим, что определения функций и значения переменных, созданные в документе или подгруженные в него из других документов, удаляются обычным образом. А именно: определения пользовательских функций и макросы удаляются функциями `remfunction` и `kill`; определения объявленных массивов, динамических массивов и массивовых функций удаляются функциями `remaarray` и `kill`; значения простых переменных, списков, матриц, элементов объявленных и динамических массивов, элементов массивовых функций удаляются функциями `remvalue` и `kill`. При выполнении функций удаления их аргументы не вычисляются. По функциям `remfunction`, `remaarray` и `remvalue` возвращается список удаленных объектов. Причем, при удалении отсутствующего объекта или объекта не того типа на соответствующее место в списке помещается слово *false*. По функции `kill` всегда возвращается слово *done* (сделано).

## СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

- [1] Есаян, А. Р. Maxima. Данные и графика. / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин. – Тула: Изд-во Тул. гос. пед. ун-та им. Л. Н. Толстого, 2011 – 367 с..

Тульский государственный педагогический университет им. Л. Н. Толстого  
Поступило 8.11.11