

## Re: limits at infinity Re: Maple bug the long liver, 1992--2007--? (limit)

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math.symbolic/2007-06/msg00056.html>

---

- *From:* "Richard J. Fateman" <[fateman@xxxxxxxxxxxxxxxxxxxxx](mailto:fateman@xxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 15 Jun 2007 13:45:19 -0700
- 

There are several research question in this bug or not bug. They have been voiced from time to time. To my knowledge it was of concern to the Macsyma project at least as early as 1970. I probably proposed to study it in some (unfunded) proposal to the NSF, within the last decade.

Before doing a (symbolic, or other) calculation, try to estimate the effort to complete it. This is, in general, impossible. But in specific instances, sometimes easy. Without computing any coefficients, you can deduce a close estimate for the number of coefficients in the answer to a dense polynomial multiplication.

In the course of a (symbolic, or other) calculation, have the computer be sufficiently "aware" of what is going on so that it halts ---before--- running out of some vital resource and crashing. Often this takes the form of stack overflow. It is, in some contexts, a bad thing to check for stack overflow. Specifically when the doofus who is comparing programs only compares their TIMES. In other contexts of course this is totally absurd. E.g. You could build an airplane that weighed less and cost less, and yet transport the same number of passengers. Leave off the safety systems.

I have, in specific instances, been told that the Lisp system we built at Berkeley (circa 1978) was too slow. After all, we checked for stack overflows. This makes sense only if you can a prior prove that the amount of stack that you provide is sufficient for any calculation, or alternatively, you don't care if the system crashes sometimes.

From the self-aware perspective, if a CAS is asked to do something that takes too much of some resource, it should not lead to a system crash, and so even if VB's questions are essentially "user error", it is impolite for the CAS to crash the program (or even the whole OS).

This kind of feature is not easy to build. For example, Macsyma/Maxima can be run on several different implementations of Common Lisp. Some implementations treat stack overflow as an opportunity to crash. Others pause and allow you (or some program) to increase the size or signal a (recoverable) error.

Thus the Maxima system must rely on features not described in the ANSI Common Lisp standard. It is not clear if this is also true of Maple or Mathematica's implementation base.

It is, in my view, clear that "out of memory, machine crash" is not a "feature".

RJF

Re: limits at infinity Re: Maple bug the long liver, 1992--2007--? (limit)