

Re: what's it worth to write a short program for polynomial multiplication?

Re: what's it worth to write a short program for polynomial multiplication?

Source: <http://sci.tech-archive.net/Archive/sci.math.symbolic/2008-06/msg00004.html>

- *From:* rjf <fateman@xxxxxxxx>
 - *Date:* Mon, 2 Jun 2008 14:57:48 -0700 (PDT)
-

On Jun 2, 1:04 pm, hru...@xxxxxxxxxxxxxxxxxxxxxx (Herman Rubin) wrote:

.... snip

Alas, it is unlikely that the superfast subimbecile of a computer will be able to do this at run time; even at compile time, it is likely that the user will have to tell the compiler what is provided and what is wanted.

a computer algebra system could run a program "what is a good algorithm for "*" at runtime.
in some sense, common lisp already does this for numeric types, and with CLOS it can do it for polynomials etc.

Sparse polynomials and dense polynomials should be handled differently, and it is usually the user who can provide information on what is sparse and what is dense.

Perhaps the user could, but rarely is the user willing. Often the user is not even available.

There is, I think, substantial evidence that the notion of object-oriented choice of operator based on one operand, and in the typical shallow way offered by most OO programming languages, is rather weak for helping in determining the proper algorithm.

Why should not the object-oriented choice of operator use the types of all the operands?

Re: what's it worth to write a short program for polynomial multiplication?

Good question. Most OO programming languages work that way. CLOS, the Common Lisp Object System is the one exception I am aware of, but perhaps there are more.

..... snip

This was the motivation for several programs of study in computer algebra systems, and resulted in several system designs, including most notably, Axiom/ Aldor.

I have observed that mathematicians, as well as physicists, may not be as good as they think they are in designing computer programming languages.

Has there been a real effort on this?

I think so.

Getting input from all on what is wanted, and then trying to do as well as one can.

You are welcome to try, again.

... snip..

Again, has anyone tried to produce an assembler which can handle weakly typed arguments with type overrides?

Apparently you have, but I do not know of anyone promoting assembler except as the target language for higher level languages.

I tried it on two machines, the CYBER 205 and the VAX 780, and came up with rather simple procedures. These ideas will also work on the POWER machines and related one.

What I did was to look at the machine instructions from a mathematical point of view and construct a simple grammar from the lot. Seymour Cray's assembler languages are not bad, but he did not try overloading.

Re: what's it worth to write a short program for polynomial multiplication?

Re: what's it worth to write a short program for polynomial multiplication?

Grammars for "machine architecture" have been explored, primarily to make the instruction generation component of compilers more table- or grammar- driven. I think work by Glanville..

<http://portal.acm.org/citation.cfm?id=512785&dl=ACM&coll=portal>

RJF