

## Re: Computer Algebra Algorithms lisp vs. C.

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math.symbolic/2005-04/msg00100.html>

---

- *From:* bernard parisse <[bernard.parisse@xxxxxxxxxxxxxxxxxxxxx](mailto:bernard.parisse@xxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 17 Apr 2005 22:00:31 +0200
- 

>>I mean that you can use the same programming language  
>>for CAS algorithms than for what is called kernel programming  
>>in many CAS.  
>  
>  
> You can, but your code will be orders of magnitude bigger, slower,  
harder to  
> use, less maintainable and less future proof. More importantly, the  
> resulting code will be dependent upon the underlying data structures.  
>

I disagree again. There are much more chances that code written  
in a non-standard language like maple language might not work  
with a future release of Maple than C++ (or Lisp, or any  
standard language) code. Moreover, people using standard  
language tend to use powerful development environment not  
available in non-standard language, there are more chances  
that they comment their code appropriately.

>  
>>>If you have a free compiler/interpreter then writing programs in a  
>>>different language does not make them inaccessible. Indeed, it makes  
them  
>>>more accessible.  
>>  
>>No, because you can not use them easily inside another program.  
>  
>  
> Just supply the external program with a string.  
>

That means parsing and printing each time you want to call a CAS function  
which is fairly inefficient. Imagine you do that with  
multivariate polynomials which tend to print to 1Mb size of text  
even with a few variables.

>

## Re: Computer Algebra Algorithms lisp vs. C.

>>He chooses C++ as implementation language because that is the  
>>natural choice given the non-CAS components.  
>  
>  
> Most people would not choose C++, of course.  
>

If you are implementing a GUI, there are great chances that you will use C++.

>  
>>If he has access  
>>to a C++ CAS library, it is much more accessible for him than  
>>if he has to exchange data from C++ types to the "host language".  
>  
>  
> In the general case (most external programs, most external languages) users  
> will not want to interface to template C++ code.  
>

I was speaking of someone writing a C++ program.  
>  
>>The host language is for me mainly a scripting language.  
>  
>  
> I don't think that statement means anything.  
>

I think it is clear for everyone administrating a Unix station.  
You do not expect the same thing from a scripting language  
than from a language like C.

>  
>>>So you're saying that the advantage of C++ is that it is the "standard  
>>>language"?  
>>  
>>Not *the* standard language, but *a* standard language.  
>>Unlike maple language which is only used by maple, mupad language  
>>by mupad, mathematica language by mathematica, etc.  
>  
>  
> Yes, exactly. C++ is one of many standard languages. So there is no  
standard  
> language, so there is no point in arbitrarily choosing C++ and then  
calling  
> it "standard". You might as well choose any language. In which case,

## Re: Computer Algebra Algorithms lisp vs. C.

choose

> the most appropriate language which, in this case, is definitely not C++.

>

You did not give any argument that support this claim. I claim that there is no "most appropriate language", it will depend on other factors and C++ is an appropriate language, I support my claim by the existing C++ CAS libraries (e.g. my giac/xcas project). Of course Lisp, Ocaml, etc. are certainly appropriate languages too, there are CAS system written in Lisp, I don't know of any Ocaml CAS system (I mean comparable to maxima, axiom or xcas), maybe someone will develop one, we'll see...

OK, I think I've lost enough time on this thread, I will most probably not convince people who are absolutely certain that C++ is not a good language for CAS algorithms. I hope that people who know C++ will just try C++ CAS libraries and make their opinion on the code, not on unsupported claims.

Bye

.