
Subject: Re: REDUCE

Posted by [John-David T. Smith](#) on Fri, 30 Mar 2001 05:32:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Kenneth Mankoff wrote:

>
>> The question, to all you C-programmers: is there a better way?
> [snip]
>> ...the code logic to compute the maximum will be the same, both
>> symbolically for all types for many types, in the compiled code itself.
>
> Hi JD,
>
> hmmm... not 100% sure, but wouldn't c++ templates solve this problem?
>
> And for the cases where it is "symbolically" the same but not "compiled
> the same", I'm not sure what this means, but I'm guessing you would handle
> these cases with overloading your operators.
>
> Of course, C isn't C++, so this might not help.
>
> I can provide code examples and more info if you wish.

Thanks for the suggestion. I had thought of that option, but I don't know much about templates, nor about linking C++ to IDL. I wonder whether the templates are just similar to my super macro for creating a different version for each type. Can you frame the maximum function I suggested in terms of a skeleton template which would operate on all the data types?

My comment with respect to compiled and symbolic maybe wasn't clear. I really just meant that you have this same code replicated over and over, with minor changes in the types of the variables used, but otherwise logically and symbolically intact. I can imagine the compiler emitting different code for, e.g., multiplying two integers, vs. two floats, but I can also imagine other types where the codes emitted are exactly the same. Obviously, you can't get something for nothing, but if real repetition exists within the compiled code, you should be able to eliminate it somehow.

Thanks again,

JD
