

CoCoALib - Bug #818

F5 with 3 args?

23 Nov 2015 21:26 - John Abbott

Status:	In Progress	Start date:	23 Nov 2015
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	10%
Category:	Tidying	Estimated time:	0.00 hour
Target version:	CoCoALib-1.0	Spent time:	3.10 hours
Description			
<p>In TmpF5.H two fns called F5 are declared: one (inline) with 2 args, and one with 3 args (3rd arg has a default value). The inline fn just calls the other one with 2 args (exploiting the presence of the default value for the 3rd arg).</p> <p>In TmpF5.C there is a defn of 2 arg fn called F5; the types match the types of the first 2 args in the 3 arg fn declared in TmpF5.H.</p> <p>test-F5.C there is a call to the inlined 2 arg version of F5. Somehow this call causes GlobalManager to report "Imminent disaster" during its destructor -- odd!</p> <p>What I want to know is how this code ever compiled???</p> <p>Anyone got any ideas?</p>			

History

#1 - 23 Nov 2015 21:28 - John Abbott

- Category set to Tidying
- Target version set to CoCoALib-1.0

If Arri comes my way I'll shorten his life by at least 2 hours -- the time I've wasted tracking this bug down. Grrr!

Since the code compiles (without warning even) even with recent versions of g++ I can only suppose that it is somehow valid... I'm amazed, and also (negatively) shocked.

#2 - 23 Nov 2015 21:31 - John Abbott

An easy workaround is just to comment out the 3rd arg in the declaration.

Since no one ever calls it with a 3rd arg (hopefully **that** would not compile!), I see no problems in doing this.

I can see an imminent issue pushing for the removal of Arri's code!

#3 - 23 Nov 2015 21:34 - John Abbott

- Status changed from New to In Progress
- % Done changed from 0 to 10

So why did this problem not appear earlier? And why did it appear now?

#4 - 23 Nov 2015 21:57 - John Abbott

With 3rd formal arg commented out, it compiles and runs fine, but produces a different answer from before... :-/ At least no errors (or other nasty surprises) appear.

Maybe I'll just disable the test.

#5 - 23 Nov 2015 22:18 - John Abbott

Arri's life is getting shorter by the minute -- grrr!

There is a 3 arg definition in TmpF5Mat.C:867, but the 3rd arg is ignored.

He uses a couple of globals, one of which is a (raw!) pointer to a PPMonoid.

I see little chance of getting this code into any decent shape...

Disabling the test test-F5 as I'm fed up with wasting my time on this code.

#6 - 24 Nov 2015 10:11 - John Abbott

- Assignee set to John Abbott

I awoke this morning with a plausible explanation of the problems with test-F5.

The problem is that the dtor for GlobalManager reports that the ref count for RingQQ is wrong. As mentioned earlier there is a (raw) point to a PPMonoid; now each PPMonoid contains a PPOrdering, and I have just changed PPOrdering so that it contains a proper ordering matrix, whereas previously the function myOrdMatCopy would generate the matrix (in someone else's space) on the fly. The contained matrix has a reference to its ring!

Anyway, I'm pretty sure that Arri's code leaks at least one PPMonoid (perhaps even one per call?) through that global raw pointer.