

CoCoALib - Bug #631

Ambiguous: rank for matrix (in ex-matrix1.C)

26 Sep 2014 15:40 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	26 Sep 2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Portability	<b>Estimated time:</b>	3.40 hours
<b>Target version:</b>	CoCoALib-0.99536 June 2015	<b>Spent time:</b>	3.50 hours
<b>Description</b>			
Apple LLVM version 6.0 (Laura's new computer) reports that rank is ambiguous in ex-matrix1.C			
<b>Related issues:</b>			
Related to CoCoALib - Design #619: Modulus (for CRTMill) ambiguous		<b>Closed</b>	<b>09 Sep 2014</b>
Related to CoCoALib - Feature #639: Shadow CoCoA namespace to help guarantee ...		<b>New</b>	<b>28 Oct 2014</b>
Related to CoCoALib - Design #642: Move code in test file into namespace CoCoA		<b>Closed</b>	<b>29 Oct 2014</b>

History

#1 - 14 Oct 2014 18:23 - John Abbott

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

I think the problem arises because rank(M) entails a user type conversion from matrix to ConstMatrixView.

std::rank is apparently new in C++11.

I see two ways around this: change name of the rank function (e.g. simply rk), or change defn of rank so that if called with a matrix it does not need the user type conversion (e.g. add overload rank(const matrix& M))

#2 - 14 Oct 2014 18:26 - Anna Maria Bigatti

John Abbott wrote:

I think the problem arises because rank(M) entails a user type conversion from matrix to ConstMatrixView.

std::rank is apparently new in C++11.

I see two ways around this: change name of the rank function (e.g. simply rk), or change defn of rank so that if called with a matrix it does not need the user type conversion (e.g. add overload rank(const matrix& M))

rk is actually a pretty standard notation, like deg or det.  
I guess we used the full word rank because it's short enough ;-)

(On the other hand, maybe the overloading is safer?)

### #3 - 14 Oct 2014 20:03 - John Abbott

- Assignee set to John Abbott

- % Done changed from 10 to 50

After speaking to Anna by phone, I have changed the name of rank into rk.

I'm ready to do CVS check in, but first I want to know what Anna thinks about the names myRank and RankByGauss. I have not shortened them because I think the full version is more comprehensible; but if we keep the longer names then we are not consistent in our use of the abbreviated form rk.

What do you think about myRk and RkByGauss?

### #4 - 14 Oct 2014 21:25 - John Abbott

Strictly speaking the proper solution to the problem is to put a using CoCoA::rank; inside the function which will call rank. As far as I can tell this is a "future-proof" solution in that it should work correctly regardless of changes that may be made to namespace std in the future.

I tried simply doing using namespace CoCoA; instead of using CoCoA::rank; but that did not remove the ambiguity (according to my compiler).

I have definite doubts about how sane/maintainable/comprehensible it would be to put using commands for all CoCoALib fns which appear in a function.

### #5 - 14 Oct 2014 21:32 - John Abbott

I now believe the ambiguity would disappear if the entire function were placed inside namespace CoCoA. We could modify all examples so that they are inside namespace CoCoA...

### #6 - 15 Oct 2014 12:06 - Anna Maria Bigatti

John Abbott wrote:

I now believe the ambiguity would disappear if the entire function were placed inside namespace CoCoA. We could modify all examples so that they are inside namespace CoCoA...

Interesting to know, I'll make my own tests also on Linux and MinGW.  
For this particular case I think the best solution would be renaming it as rk.

### #7 - 15 Oct 2014 12:13 - Anna Maria Bigatti

John Abbott wrote:

After speaking to Anna by phone, I have changed the name of rank into rk.

I'm ready to do CVS check in, but first I want to know what Anna thinks about the names myRank and RankByGauss. I have not shortened them because I think the full version is more comprehensible; but if we keep the longer names then we are not consistent in our use of the abbreviated form rk.

What do you think about myRk and RkByGauss?

hmmm.... undecided.....

yes, I would go for myRk and RkByGauss.

**#8 - 27 Oct 2014 10:03 - John Abbott**

In comment 5 I suggesting putting all code in namespace CoCoA; perhaps we should try doing this for some CoCoALib tests first (so we can see if it works as expected, and how it looks). Assuming the trial goes well, I think it would probably be a good idea to do it for all tests. Afterwards we can decide whether it is a good idea also for the examples.

**UPDATE** this suggestion has become issue [#642](#)

**#9 - 29 Oct 2014 11:48 - John Abbott**

- *Subject changed from Ambiguous: rank for matrix (in ex-matrix1.C to Ambiguous: rank for matrix (in ex-matrix1.C)*

**#10 - 20 Apr 2015 12:34 - Anna Maria Bigatti**

Renaming is done now. Isn't it?

**#11 - 11 May 2015 15:32 - John Abbott**

- *Status changed from In Progress to Feedback*

- *% Done changed from 50 to 90*

I think everything has now been done.

**#12 - 11 May 2015 16:02 - John Abbott**

I noticed that some of Christof's code uses rank; I have written to him asking whether it would be reasonable to change his fn rank into rk. The C5 interface also uses a RECORD with a field called Rank. Waiting for a reply from Christof.

**#13 - 25 Jun 2015 15:27 - John Abbott**

JAA notes that now we're planning to move all test/example code into namespace CoCoA the problem of ambiguity (in names such as rank) goes away -- the positioning inside the namespace resolves the issue.

Anna replies that it is not so convenient for a user to have to place their code inside the CoCoA namespace - indeed this could be a problem with code which uses several libraries.

Anna also suggest offering both **rk** and **rank**. JAA is uncertain about "having two names for the same function".

Comments? Opinions? Suggestions?

**#14 - 30 Jun 2015 15:07 - John Abbott**

- *Status changed from Feedback to Closed*

- % Done changed from 90 to 100
- Estimated time set to 3.40 h

We use the name **rk** for the main function, but allow Rank as a component of a compound name (to increase readability).