CoCoALib - Bug #814

PPOrdering: matrix ordering, what rings are allowed.

23 Nov 2015 11:49 - John Abbott

Status:	Closed	**Start date:**	23 Nov 2015
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Safety	**Estimated time:**	2.01 hours
Target version:	CoCoALib-0.99550 spring 2017	**Spent time:**	1.60 hour

Description

I think that NewMatrixOrdering does not check the ring of the matrix handed to it.

Check if I am right; anyway, make sure that it is evident from the code that a suitable check is made.

Related issues:

Related to CoCoALib - Design #827: NewPositiveMat also for matrices over QQ? ...	**Closed**	**26 Nov 2015**
Related to CoCoALib - Bug #820: NewMatMinimize, NewMatCompleteOrd - a godfors...	**Closed**	**25 Nov 2015**
Related to CoCoALib - Design #707: MatrixOrderingMod32749Impl: test and write...	**In Progress**	**15 May 2015**

History

#1 - 25 Nov 2015 13:34 - John Abbott

The following example compiles and runs, and prints out a matrix with rational entries.

```
matrix M = NewDenseMat(RingQQ(),2,2);
SetEntry(M,0,0,BigRat(1,2));
SetEntry(M,0,1,BigRat(1,1));
SetEntry(M,1,0,BigRat(3,2));
SetEntry(M,1,1,BigRat(2,1));

PPOrdering ord = NewMatrixOrdering(M,1);
cout << ord << endl;
```

I think the fn should accept a matrix over any ring (of char 0) whose entries are integer; internally it should store the matrix as a matrix over ZZ.

What do you think?

#2 - 25 Nov 2015 13:37 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 10

If the input matrix does not have integer entries then the pseudo-ctor should give error (it may delegate the task to the real ctor).

While it is always possible to scale a matrix of rationals so that all entries are integer, I don't think this should be done automatically.

#3 - 25 Nov 2015 14:30 - John Abbott

I have added the check to the ctor for MatrixOrderingImpl, and also added a new test test-bug5.C. I'll check these in with the next batch.

#4 - 25 Nov 2015 14:32 - John Abbott

There are 2 pseudo-ctors for NewMatrixOrdering:

```
PPOrdering NewMatrixOrdering(long NumIndets, long GradingDim, ConstMatrixView OrderMatrix);
PPOrdering NewMatrixOrdering(ConstMatrixView OrderMatrix, long GradingDim);
```

The order of their args is inconsistent. What order do we want?

#5 - 26 Nov 2015 13:08 - John Abbott

Now I wonder why there is the first version of the pseudo-ctor:

```
PPOrdering NewMatrixOrdering(long NumIndets, long GradingDim, ConstMatrixView OrderMatrix);
```

The value of NumIndets must be equal to NumCols(OrderMatrix), so why do we ask the user to specify it twice?

#6 - 26 Nov 2015 13:14 - John Abbott

Could it be useful to have a pseudo-ctor which accepts just the "grading matrix"? The pseudo-ctor would then "auto-complete" the matrix to a term order (presumably by appending StdDegRevLex).

If we do adopt this idea, what are the semantics? Which do you think is better?

```
NewMatrixOrdering(GrdMat);
NewMatrixOrdering(CompleteToOrd(GrdMat));
```

The second is longer but more explicit...
Perhaps I have a mild preference for the second -- I'm often suspicious of automatisms. Also, if we adopt initially the semantics of requiring a full order matrix, and then decide later that it would be nice to accept just the "grading part" then it is easy to extend the semantics without breaking existing (correct) code.

#7 - 01 Dec 2015 15:39 - John Abbott

Refering to comment 4: I have now removed the signature with 3 args (after speaking to Anna about it).

#8 - 24 Mar 2016 12:06 - Anna Maria Bigatti

- Related to Design #827: NewPositiveMat also for matrices over QQ? Also NewIntegerOrdMat. (now called MakeTermOrd) added

#9 - 24 Mar 2016 12:16 - Anna Maria Bigatti

John Abbott wrote:

> Could it be useful to have a pseudo-ctor which accepts just the "grading matrix"? The pseudo-ctor would then "auto-complete" the matrix to a term order (presumably by appending StdDegRevLex).

I think "no"

> If we do adopt this idea, what are the semantics? Which do you think is better?
> [...]

I think it is ambiguous what the GradingDim should be... 0? 1? NumRows(M)?
All these choices make sense in some context (i.e. for some user)

#10 - 24 Mar 2016 18:10 - Anna Maria Bigatti

- Related to Bug #820: NewMatMinimize, NewMatCompleteOrd - a godforsaken mess! added

#11 - 24 Mar 2016 18:13 - Anna Maria Bigatti

- Related to Design #707: MatrixOrderingMod32749Impl: test and write documentation! added

#12 - 16 Jun 2016 17:14 - John Abbott

- Target version changed from CoCoALib-0.99540 Feb 2016 to CoCoALib-0.99550 spring 2017

What is the answer to the original question?

Comment 1 proposed an answer, and asked for discussion. The other comments suggest something has been implemented, but no explicit answer is given here.

I suppose that this issue can be closed now (or very soon).

#13 - 16 Sep 2016 18:08 - John Abbott

- Status changed from In Progress to Closed

- % Done changed from 10 to 100

The implementation now includes an **implicit** check that the ring of the given matrix is char 0; internally it simply calls ConvertTo<BigInt>(matrix(i,j)) which effectively does the check.

This check is performed in the ctor for MatrixOrderingImpl.

The internally stored matrix has entries in RingZZ().

The documentation for NewMatrixOrdering has been updated.

Closing.

#14 - 28 Apr 2017 09:33 - Anna Maria Bigatti

- Estimated time set to 2.01 h