

CoCoALib - Bug #820

NewMatMinimize, NewMatCompleteOrd - a godforsaken mess!

25 Nov 2015 15:45 - John Abbott

Status:	Closed	Start date:	25 Nov 2015
Priority:	High	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Tidying	Estimated time:	0.00 hour
Target version:	CoCoALib-0.99540 Feb 2016	Spent time:	1.55 hour
Description			
JAA really thinks that order matrices should be over ZZ rather than QQ since we insist that the entries be integer (otherwise WDeg needs to be changed).			
The appalling mess that comprises the fns used for "completing" matrices into terms orders needs a radical overhaul.			
Related issues:			
Related to CoCoALib - Bug #834: Fix test failures (after revising MatrixForOr...		Closed	09 Dec 2015
Related to CoCoALib - Design #822: Should ElimMat return a ConstMatrix		Closed	25 Nov 2015
Related to CoCoALib - Design #827: NewPositiveMat also for matrices over QQ? ...		Closed	26 Nov 2015
Related to CoCoALib - Bug #814: PPOrdering: matrix ordering, what rings are a...		Closed	23 Nov 2015

History

#1 - 26 Nov 2015 14:14 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I have hacked the code: assuming all matrices are over ZZ makes the code a lot simpler. I have added a fn which creates an equivalent matrix over ZZ if the input matrix is not over ZZ; publicly visible fns call this if necessary on their args.

#2 - 26 Nov 2015 14:53 - John Abbott

What exactly should NewMatCompleteOrd do?

What does its expect as input? And what does it give as output? When does it give error?

From its name I would expect it to produce a term ordering "compatible" with the input matrix. Possibly the first rows of the resulting matrix must be equal to the input matrix.

What should NewMatCompleteOrd(ZeroMat(...)) do?

This might give error since it is not possible to produce a full-rank square matrix having first row full of zeroes -- if we adopt the semantics that the first rows must match. Currently this produces a deg-rev-lex matrix; is that right?

#3 - 26 Nov 2015 14:58 - John Abbott

- Description updated

The CoCoALib name is NewMatCompleteToOrd while in CoCoA-5 it is CompleteToOrd.

Is this what we want?

#4 - 26 Nov 2015 15:07 - John Abbott

I suggest that the semantics of `NewMatCompleteToOrd` be the following:

- expect as input a matrix M (over ring of char 0, whose entries are integers) of full (row) rank where each col is either zero or the first non-zero element is positive
- result is a (full rank) square matrix defining a term ordering whose first rows are equal to those of M

The fn gives error if the input conds are not satisfied.

Comments? Criticisms? Other ideas?

#5 - 26 Nov 2015 18:06 - Anna Maria Bigatti

John Abbott wrote:

I suggest that the semantics of `NewMatCompleteToOrd` be the following:

yes, I agree.

I'm just a bit undecided about the `QQ/ZZ` choice: originally it was `ZZ`, but then I changed the code so that it would accept `QQ` (and the rank is indeed computed in `QQ`). I'm a bit concerned about changing back..

#6 - 26 Nov 2015 18:09 - Anna Maria Bigatti

John Abbott wrote:

The `CoCoALib` name is `NewMatCompleteToOrd` while in `CoCoA-5` it is `CompleteToOrd`.
Is this what we want?

I was following the `cocoalib` rule of "New" when creating a new matrix, but I didn't like it in `cocoa-5`.
... but you are right: that's not nice.

#7 - 26 Nov 2015 18:40 - John Abbott

I am pretty convinced that `ZZ` is the right ring for order matrices in `CoCoALib`.

That said, I am making sure that the functions accept matrices over any ring of char 0, but the entries must be integer. If you ask a poly ring (or `PPMonoid` for its order matrix, what you get back is a matrix over `ZZ`).

Not having a matrix over `QQ` is a nuisance if you want to compute its inverse, but it is fairly easy to convert it to a matrix over `QQ` by using apply with an embedding homomorphism (`ZZ --> QQ`).

I also had to change some tests which compared the result of `OrdMat` to a given matrix (which by default is over `QQ`); I felt that the changes were not too burdensome, though it was nicer not to have to specify the ring... but that's a consequence of our "unnatural choice" that a matrix constructed

explicitly from a list of lists of integers is taken to be over QQ...

#8 - 01 Dec 2015 15:34 - John Abbott

I shall shortly try to implement the semantics given in comment 4 above.

After speaking to Anna, the idea is that only `NewMatCompleteToOrd` will remain publicly visible; the other fns will either be removed or hidden from public access.

For the time being the precise definition of the single function will be deliberately a bit vague, so we have some leeway in completing a matrix to an ordering.

#9 - 04 Dec 2015 15:03 - John Abbott

- Assignee set to John Abbott

I do not much like the names `CompleteToOrd` and `NewMatCompleteOrd`.

Perhaps `MakeTermOrd` or `MakePositiveTermOrdering`, or something similar?

Comments? Ideas? Suggestions?

#10 - 04 Dec 2015 15:11 - Anna Maria Bigatti

John Abbott wrote:

Perhaps `MakeTermOrd` or `MakePositiveTermOrdering`, or something similar?

`MakeTermOrd`!

#11 - 24 Mar 2016 18:01 - Anna Maria Bigatti

- Related to Design #822: Should `ElimMat` return a `ConstMatrix` added

#12 - 24 Mar 2016 18:08 - Anna Maria Bigatti

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

#13 - 24 Mar 2016 18:09 - Anna Maria Bigatti

- Subject changed from `NewMatMinimize`, `NewMatCompleteOrd`, `AuxMAndElim`, `AuxElimandM`, `ElimRow` - a godforsaken mess! to `NewMatMinimize`, `NewMatCompleteOrd` - a godforsaken mess!

- Description updated

- Status changed from `In Progress` to `Closed`

- % Done changed from 10 to 100

#14 - 24 Mar 2016 18:10 - Anna Maria Bigatti

- Related to Bug #814: `PPOrdering`: matrix ordering, what rings are allowed. added