

## CoCoALib - Design #602

### OrdMat: should it be a reference to a MatrixView in all PPOrderings?

31 Jul 2014 17:47 - Anna Maria Bigatti

<b>Status:</b>	Closed	<b>Start date:</b>	31 Jul 2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	New Function	<b>Estimated time:</b>	5.50 hours
<b>Target version:</b>	CoCoALib-0.99540 Feb 2016	<b>Spent time:</b>	5.20 hours
<b>Description</b>			
The function <b>OrdMat</b> needs to create a new copy of the ordering matrix because not all PPOrdering objects actually contain a matrix (e.g. lex does not). We could store a <b>MatrixView</b> in each PPOrdering.			
<b>Related issues:</b>			
Related to CoCoALib - Feature #310: ordering and grading (weights) matrix		<b>Closed</b>	<b>13 Feb 2013</b>
Related to CoCoALib - Feature #803: PPOrdering: use it to compute WDeg?		<b>In Progress</b>	<b>11 Nov 2015</b>

#### History

##### #1 - 11 Nov 2015 13:35 - John Abbott

- Description updated

How important is it that OrdMat is fast? Is there a good example where we would want to compute repeatedly the OrdMat for an ordering? [JAA has doubts]

Remember KISS!

##### #2 - 11 Nov 2015 16:35 - John Abbott

JAA now thinks that there is an example (see issue [#803](#)) where it could be useful to have the order matrix(view) quickly available from a PPOrdering.

Having the order matrix readily accessible may also help simplify (a little) the impl in PPMonoidEv.C of the various concrete subclasses of CmpBase.

##### #3 - 26 Nov 2015 13:30 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99540 Feb 2016

- % Done changed from 0 to 10

JAA is now convinced that having OrdMat return a "reference" (namely a ConstMatrixView) is correct: it was relatively easy to implement and should incur no measurable run-time overhead.

I have a running impl, but it is not yet checked in because there are some other related issues still to be resolved.

One important detail is the ring over which the matrices will be. In practice we require that the entries of the matrix be integer, so it seems to make most sense for the matrices to be over ZZ. However, previously the matrices were over QQ, though JAA thinks there was no compelling *technical* reason for this.

JAA notes that some C5 tests involved checks similar to  $\text{OrdMat}(P) = \text{mat}([[1,2],[3,4]])$  and in C5 a matrix created from INT/RAT values is automatically over QQ even if all the values are actually integer -- this was to make interactive use simpler e.g.  $\text{inverse}(\text{mat}([[1,2],[3,4]]))$  fails if the matrix is over ZZ.

Making OrdMat return a matrix over ZZ means the tests have to be modified slightly: e.g.  $\text{OrdMat}(P) = \text{mat}(\text{ZZ},[[1,2],[3,4]])$ . JAA thinks the extra onus of having to specify ZZ explicitly in the tests is quite acceptable.

Anna, what do you think?

**#4 - 23 Mar 2016 15:24 - John Abbott**

- *Status changed from In Progress to Closed*
- *% Done changed from 10 to 100*
- *Estimated time changed from 6.00 h to 5.50 h*