CoCoA-5 - Feature #309

(Multi)BlockMatrix

13 Feb 2013 08:57 - Lorenzo Robbiano

Status:	Closed	**Start date:**	13 Feb 2013
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	CoCoA-5 function: new	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.0.3	**Spent time:**	4.40 hours

Description

It would be VERY HELPFUL to have a function, say MultiBlockMat, which takes several matrices in input and returns a matrix built up with the given matrices.
Let me explain better with examples.
Ex1. A:=Mat(QQ, [[1,2, 3], [4,5,6]]); B:= Mat(QQ, [[7,8, 9]]);
MultiBlockMat(2,1,A,B) should return Mat(QQ, [[1,2, 3], [4,5,6],[7,8, 9]]);
Ex2. A:=Mat(QQ, [[1,2, 3], [4,5,6]]); B:=Mat(QQ, [[1], [2]]); C:= Mat(QQ, [[3], [4]);
MultiBlockMat(1,3,A,B, C) should return Mat(QQ, [[1,2, 3,1,3], [4,5,6,2,4]]);
Ex3. A1:=Mat(QQ,[[1]]); A2:=Mat(QQ,[[2]]); A3:=Mat(QQ,[[3]]); A4:=Mat(QQ,[[4]]); A5:=Mat(QQ,[[5]]); A6:=Mat(QQ,[[6]]);
MultiBlockMat(2,3, A1,A2, A3, A4, A5, A6) should return Mat(QQ, [[1,2, 3], [4,5,6]]);
while
MultiBlockMat(3,2, A1,A2, A3, A4, A5, A6) should return Mat(QQ, [[1,2], [3,4], [5,6]]);

The current BlockMat is a special case, so we could also call the new function BlockMat and delete the old one.

AMB: 2013-02-13 implemented in CoCoA-5 **ConcatHorList**, **ConcatVerList**.
AMB: 2013-05-31 renamed **BlockMat** into **BlockMat2x2**
and added **BlockMat**: BlockMat([[A1,A2, A3], [A4, A5, A6]])

Related issues:

Related to CoCoALib - Feature #37: matrix constructors	**Closed**	**02 Nov 2011**
Related to CoCoA-5 - Feature #78: Elim ordering and matrix ordering in CoCoA-5	**In Progress**	**16 Jan 2012**
Related to CoCoA-5 - Feature #7: Automatic mapping between (some) rings	**Resolved**	**20 Oct 2011**

History

#1 - 13 Feb 2013 09:17 - Anna Maria Bigatti

We have this constructor for matrices:

```
/**/  MakeMatByRows(2, 10, 1..20);
matrix([
  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
  [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
])
```

Maybe we could make **MakeMultiMatByRows(2, 10, L);**
with L list of MATRIX with 20 elements.
Unfortunately the general implementation is going quite tricky (and, in CoCoA-5, costly).
Should we make it in CoCoALib (tricky, but not costly ;-)?
In fact it could be handy for elim orderings.

The usual question: a good name? **MakeMultiMatByRows**? **MakeMatByRows**?

#2 - 13 Feb 2013 14:31 - Anna Maria Bigatti

I thought about the code for the multi block matrix.
The input checking (size compatibility) would be **really tedious**.
I think something like this might do most of the desired work

```
Define ConcatHorList(L)
  If L=[] Then Error("Empty list"); EndIf;
  If Len(L)=1 Then Return L[1]; EndIf;
  Return ConcatHor(L[1], ConcatHorList(Tail(L)));
EndDefine; -- ConcatHorList

-- Example 2:
/**/ A:=Mat(QQ, [[1,2,3], [4,5,6]]); B:=ColMat(QQ, [1,2]); C:= ColMat(QQ, [3,4]);
/**/ ConcatHorList([A,B,C]);
matrix( /*QQ*/
 [[1, 2, 3, 1, 3],
  [4, 5, 6, 2, 4]]
 )

-- Example 3;
/**/ A1:=Mat(QQ,[ [1]]); A2:=Mat(QQ,[ [2]]); A3:=Mat(QQ,[ [3]]);
/**/ A4:=Mat(QQ,[ [4]]); A5:=Mat(QQ,[ [5]]); A6:=Mat(QQ,[ [6]]);
/**/ ConcatVer(ConcatHorList([A1,A2,A3]), ConcatHorList([A4,A5,A6]));
matrix( /*QQ*/
 [[1, 2, 3],
  [4, 5, 6]]
 )
```

#3 - 13 Feb 2013 16:28 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

- Target version set to CoCoA-5.0.3

- % Done changed from 0 to 80

added ConcatHorList, ConcatVerList and manual

#4 - 13 Feb 2013 16:37 - John Abbott

JAA wonders whether it might be appropriate to have a fairly general function for building matrices by blocks:

```
BlockMat(M)  --->  M
```

```
BlockMat([M1, M2, M3])  --->  ConcatHorList(M1,M2,M3)
BlockMat([[M1,M2,M3],[M4,M5,M6]])  --->  ConcatVerList([ConcatHorList(row) | row in L])
```

What do you think?

#5 - 13 Feb 2013 17:04 - Anna Maria Bigatti

John Abbott wrote:

> JAA wonders whether it might be appropriate to have a fairly general function for building matrices by blocks:

That's what I thought to start with, but the current solution is actually more flexible (in philosophy). I explain: I would expect **BlockMat** to check row and column numbers compatibility (**very tedious**) whether

```
ConcatVerList([ConcatHorList(row) | row in L])
```

would check compatibility for all row numbers, but only for **total column number**, e.g.

```
A := RowMat([1,2,3]);  B := RowMat([0,0]);
/**/ BlockMat(A,B, B,A);  -- gives error, as it should!
/**/ ConcatVerList([ConcatHorList([A,B]),  ConcatHorList([B,A]) ]);
matrix( /*QQ*/
 [[1, 2, 3, 0, 0],
  [0, 0, 1, 2, 3]]
 )
```

... maybe I could add your definition to the manual of **BlockMat** calling it **MultiBlockMat** and let the user use it without the checks ;-)

#6 - 13 Feb 2013 18:11 - Anna Maria Bigatti

added JAA's definition of MultiBlockMat into manual entry of BlockMat
(and also search keys)

```
  Define MultiBlockMat(L)
    Return ConcatVerList([ConcatHorList(row) | row in L]);
  EndDefine;

/**/  A := RowMat([1,2,3,4]);   B := RowMat([0,0]);
/**/  BlockMat(A,B, B,A);   -- !!! ERROR !!! as expected
/**/  MultiBlockMat([[A,B], [B,A]]);
matrix( /*QQ*/
 [[1, 2, 3, 4, 0, 0],
  [0, 0, 1, 2, 3, 4]]
 )
```

#7 - 14 Feb 2013 11:10 - Anna Maria Bigatti

- Status changed from New to Feedback

- % Done changed from 80 to 90

#8 - 14 Feb 2013 15:48 - John Abbott

How about renaming the current BlockMat to **BlockMat2x2**,
and the new fn MultiBlockMat to just plain **BlockMat**?

Just so we don't forget: for the time being we are **not planning** on making a MultiBlockMat fn **available in CoCoALib** -- it would be tedious making up the vector of vector of matrix input data.

#9 - 15 Feb 2013 11:31 - John Abbott

If in the future we let users be more flexible with rings, it may be helpful to have a BlockMat function which accepts also the ring in which the entries will reside -- the entries of the contributing matrices will be mapped automatically into that ring.

The syntax I have in mind is something like

```
BlockMat(R, [[Mat1, Mat2],[Mat3, Mat4]]);
```

This syntax appears to be coherent with other matrix ctors. Similar changes might then be needed to ConcatVer and ConcatHor, and so on.

Is this a good/sane idea?

#10 - 31 May 2013 09:32 - Anna Maria Bigatti

- Status changed from Feedback to Closed

Final decision is to make it public because it is indeed quite handy
(but has an uneven behaviour on rows and cols and it's not worth the effort in making it much cleaner).
Fixed the documentation.

#11 - 31 May 2013 09:34 - Anna Maria Bigatti

- % Done changed from 90 to 100

#12 - 31 May 2013 10:30 - John Abbott

JAA still prefers the name **BlockMat** to the unwieldy MultiBlockMat.
New issue for 5.0.4??

#13 - 31 May 2013 10:45 - Anna Maria Bigatti

John Abbott wrote:

> JAA still prefers the name **BlockMat** to the unwieldy MultiBlockMat.
> New issue for 5.0.4??

BlockMat does another thing and has the same syntax as in CoCoALib (one-liner).
The new one would be a nightmare in C++. So we need two different names.
(we could do CoCoA5-overloading, but that's an unnecessary complication with all the things we have to do)

#14 - 31 May 2013 14:41 - Anna Maria Bigatti

- Subject changed from MultiBlockMatrix to (Multi)BlockMatrix