

CoCoALib - Feature #1198

Non-standard DegRevLex (NonStdDegRevLex, WDegRevLex)

27 Jun 2018 15:08 - John Abbott

Status:	New	Start date:	27 Jun 2018
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	New Function	Estimated time:	0.00 hour
Target version:	CoCoALib-1.0	Spent time:	0.80 hour
Description			
There is a family of cases where a "DegRevLex" ordering with "non-standard" weights is useful. This can be implemented by a matrix ordering, but that is inefficient (esp. if there are many indets, then creating the ring can cost more than the computation in the ring).			
Maybe the PPOrdering family should be called WDegRevLex?			

History

#1 - 27 Jun 2018 15:17 - Anna Maria Bigatti

In particular we should have a dedicated OrdvArith implementation for it.

This should make exponents faster, and even the creation of the ring faster (useful for SC2 computations, with many big rings, and trivial GBases)

#2 - 02 Jul 2018 18:42 - John Abbott

Here are some thoughts about the design/impl.

If the GradingDim is 1 then we have a vector of positive "weights": W_1, W_2, \dots, W_n

The rest of the matrix is an antidiagonal of -1 with the rest being zeros.

This is term-ord equivalent to a matrix whose lower rows are just prefixes of the first row (filled with zeroes to the correct number of columns): for example

```
[[3, 4, 5],  
 [3, 4, 0],  
 [3, 0, 0]]
```

Note that "encoding" and "decoding" are simple and fast operations.

The GradingDim is greater than 1 then it is a little more complicated but not too bad.

Roughly speaking we split the matrix into "column blocks" where each block is determined by the row index where the first non-zero element appears. Each column block is then handled as for the case with GradingDim equal to 1.

#3 - 23 Sep 2019 12:44 - John Abbott

- Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-1.0