CoCoALib - Design #841

NewPolyRing: tidy up the many different versions

21 Jan 2016 13:09 - John Abbott

Status:	In Progress	**Start date:**	21 Jan 2016
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	10%
Category:	Tidying	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	2.15 hours

Description

Currently (Jan 2016) there are many different fns called NewPolyRing offering the caller the possibility to specify (or not) various aspects.

At the moment most of these fns have very similar impls which are **independent**. I noticed this because I wanted to add a default behaviour: if there are more than 20 (say) indets then the default choice is a sparse PPMonoid; however, with the current impl design I would have to change several different fns.... not ideal.

Related issues:

Related to CoCoALib - Slug #1057: Slug: Polynomial ring contructor slow with ...	**In Progress**	**04 May 2017**

History

#1 - 21 Jan 2016 13:27 - John Abbott

JAA thinks that the best design would have all NewPolyRing fns pass through a single "base" fn; so there would be just one place where the "trick" of choosing a sparse PPMonoid has to be implemented.

Currently our default values are:

- default ordering is **StdDegRevLex**
- if an OrdCtor is given then it is passed directly to underlying ring ctor (but we would simply use it to create an explicit ordering with the correct number of indets)
- default indet names are whatever the defaults are for NewPolyRing_DMPI and for NewPolyRing_DMPII; to find these out we must look in the relevant files...

JAA thinks it would better if there were a single place where the default indet names are defined; then only one place needs to be altered if we want to change the default names in the future.

NOTE JAA observes that the caller must specify the indet names when creating a PPMonoid, so there is no default there.

#2 - 21 Jan 2016 14:09 - John Abbott

I now notice that there are also 6 fns called NewPolyRing_DMPI and 5 fns called NewPolyRing_DMPII; this gives a total of 17 fns for creating (multivariate) poly rings! That is a lot of different but very similar fns!

Moreover the relationship between the various NewPolyRing_DMPI fns mirrors that between the various NewPolyRing fns; a similar comment applies to NewPolyRing_DMPII.

This is clearly not "scalable" (which probably does not matter that much here), and looks to be a good way to make future maintenance needlessly long, tedious and error-prone.

It would seem more sensible to allow the caller to specify a "flag" saying whether a specific polyring impl is desired; then there could be a single set of fns which sort out the various default values, and the common "bottom level" fn then looks at the "flag" to decide which specific sort of polyring to

create.

#3 - 23 Jan 2016 20:48 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

[after speaking to Anna]
Here is a possible design:
there is a single "very general" pseudo-ctor which is the only way to create polyrings, all other pseudo-ctors are simply convenient "short-hand" for some common cases (their impls should be just 1-2 lines, and they call the unique general pseudo-ctor to actually create the polyring with appropriate args).

The general pseudo-ctor will need args to specify the following:

- coeff ring
- number and names of indets
- PPOrdering
- which impl to use (*e.g.* "auto", "DMPI", "DMPII", "DMPClean")

NOTE (problem!) not sure how this fits in with the pseudo-ctor which has args (CoeffRing, PPM); it does not make sense to deconstruct the PPM even if reconstructing it magically rebuilds the original PPM.

#4 - 23 Jan 2016 22:21 - Anna Maria Bigatti

I have not checked the actual code, but I think we could have abstract implementations myNewPolyRing dealing with the defaults, and only one concrete function (for each concrete class) with all the arguments.
Moreover the defaults could be defined in the appropriate class files (ordering, symbol,..)

#5 - 01 Sep 2017 11:21 - John Abbott

It would be nice to finish this issue soon!

#6 - 01 Sep 2017 11:30 - Anna Maria Bigatti

We should also consider how and when optimize copying the ordering matrix.
(I need to discuss the details in person)

#7 - 01 Sep 2017 11:30 - Anna Maria Bigatti

- Related to Slug #1057: Slug: Polynomial ring contructor slow with (big) matrix ordering added