

## CoCoALib - Design #455

### Which sets of generators in an ideal?

03 Mar 2014 18:57 - Anna Maria Bigatti

<b>Status:</b>	New	<b>Start date:</b>	03 Mar 2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Various	<b>Estimated time:</b>	16.00 hours
<b>Target version:</b>	CoCoALib-1.0	<b>Spent time:</b>	0.75 hour
<b>Description</b> now we have myGens myTidyGens but in SparsePolyRing we would also need myGBasis (currently stored in myTidyGens) myMinGens myJanetBasis			
<b>Related issues:</b>			
Related to CoCoA-5 - Feature #216: Janet Bases: port function into CoCoA-5		<b>In Progress</b>	<b>02 Aug 2012</b>
Related to CoCoALib - Feature #215: Janet Bases: check and include code in Co...		<b>Closed</b>	<b>02 Aug 2012</b>
Related to CoCoA-5 - Bug #446: intersection fails with zero ideal		<b>Closed</b>	<b>20 Feb 2014</b>
Related to CoCoALib - Feature #366: function (CoCoALib) for minimal generator...		<b>Closed</b>	<b>11 Jun 2013</b>
Related to CoCoALib - Feature #399: add myHilbertSeries member field to ideal...		<b>New</b>	<b>02 Aug 2013</b>
Related to CoCoALib - Bug #130: Design problem in ideals		<b>New</b>	<b>15 Apr 2012</b>
Related to CoCoALib - Feature #400: add myJBMil member field to ideal		<b>New</b>	<b>02 Aug 2013</b>
Related to CoCoALib - Feature #665: Integrate Janet/Pommaret basis code		<b>In Progress</b>	<b>11 Feb 2015</b>
Related to CoCoALib - Design #871: Redesign ideals		<b>New</b>	<b>26 Apr 2016</b>

### History

#### #1 - 03 Mar 2014 19:04 - Anna Maria Bigatti

- Category set to Various

- Target version set to CoCoALib-0.99534 Seoul14

from issue [Bug #446](#): intersection fails with zero ideal  
John Abbott wrote

What I tried to suggest yesterday, is that an ideal could contain **two** lists of generators: one is the list the user supplied, the other is a "cleaned list" *e.g.* without zero values, maybe without duplicates, maybe made monic, perhaps put into a good order, etc. I can see one problem: which list should be used when printing the ideal? Presumably the original user-supplied list should be returned by the mem fn myGens (or whatever it is called). Mmmm, it could be confusing to have two lists... Unless the "cleaned list" is used only when calling ideal operations...???

In fact we already have such a field: it is called myTidyGens.  
So far we have used it for an **optimal** set of generators: i.e. GBasis, (and a minimal set for ideals in ZZ?)  
Even though it would make sense to rename it, I fear it might break some code if we do it now.

**#2 - 21 Mar 2014 11:18 - Anna Maria Bigatti**

I think we should have myMinGens in every ideal: a "minimal set of generators" is quite a general concept. We should then specify in which cases that has "minimal" length or not. (as for for PIDs and poly homog ideals)

Now we have "myTidyGens", but that has different meanings depending on the ring. Now I think that's probably not a good idea: in polynomial rings we need so many more....

**#3 - 09 May 2014 13:00 - Anna Maria Bigatti**

- *Estimated time set to 16.00 h*

A similar problem is storing HilbertFunction and resolution in the ideal/module

**#4 - 14 Jul 2014 15:17 - John Abbott**

- *Target version changed from CoCoALib-0.99534 Seoul14 to CoCoALib-1.0*

**#5 - 26 Apr 2016 15:10 - John Abbott**

- *Related to Design #871: Redesign ideals added*