

CoCoALib - Feature #1206

syz, SyzOfGens: which shifts for zero?

02 Aug 2018 14:54 - Anna Maria Bigatti

<b>Status:</b>	Closed	<b>Start date:</b>	02 Aug 2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	Maths Bugs	<b>Estimated time:</b>	3.00 hours
<b>Target version:</b>	CoCoALib-0.99850	<b>Spent time:</b>	3.60 hours
<b>Description</b>			
It makes to be able to compute			
<pre>/**/ use R ::= QQ[x,y,z]; /**/ I := ideal([x, y, x+y, 0]); /**/ SyzOfGens(I);</pre>			
Because the list of generators might have some meaning (e.g. partial derivatives) but syzygies are shifted with the degrees of the generators, and what's the degree of 0? If the generators are homogeneous and some are 0, these are probably meant with some particular degree. Should we try to "guess" sometimes? for example if all gens have the same degree?			
<b>2024-03</b> Final decisions: Pass the FreeModule with the desired shifts if some entry is 0. Do not implement syz0 (which should guess) Keep SyzOfGens (more convenient for CoCoA-5)			
<b>Related issues:</b>			
Related to CoCoALib - Bug #1205: SyzOfGens: bug with zero generators		<b>Closed</b>	<b>01 Aug 2018</b>
Related to CoCoA-5 - Bug #509: SyzOfGens (CoCoA-5) does not have correct shifts		<b>Closed</b>	<b>03 Apr 2014</b>
Related to CoCoALib - Design #1647: Suppress zero from ideal generators? Det...		<b>Closed</b>	<b>20 Jan 2022</b>
Related to CoCoA-5 - Feature #1072: syz: apply to ModuleElem?		<b>Rejected</b>	<b>18 May 2017</b>
Related to CoCoA-5 - Feature #1805: Add CoCoA-5 syntax: syz(<F,> list of Modu...		<b>New</b>	<b>26 Mar 2024</b>
Related to CoCoALib - Feature #598: Syzygy for modules: non-homogeneous module		<b>In Progress</b>	<b>15 Mar 2013</b>
Related to CoCoALib - Feature #1809: Make ex-syz.C		<b>New</b>	<b>28 Mar 2024</b>

History

- #1 - 02 Aug 2018 14:55 - Anna Maria Bigatti
  - Related to Bug #1205: SyzOfGens: bug with zero generators added
- #2 - 02 Aug 2018 14:55 - Anna Maria Bigatti
  - Related to Bug #509: SyzOfGens (CoCoA-5) does not have correct shifts added
- #3 - 05 Apr 2019 15:40 - John Abbott
  - Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-1.0
- #4 - 09 Feb 2024 08:31 - Anna Maria Bigatti
  - Related to Design #1647: Suppress zero from ideal generators? Detect 1 and simplify generators? added
- #5 - 09 Feb 2024 08:38 - Anna Maria Bigatti
  - Status changed from New to In Progress
  - Target version changed from CoCoALib-1.0 to CoCoALib-0.99850
  - % Done changed from 0 to 10

Currently we have decided to forbid generator 0 in ideals ([#1647](#)), so SyzOfGens will never find a 0, keeping in mind that gens(l) might be different from the list given in  $I := \text{ideal}(L)$ .

We can arrange syz to work with 0s (they were dealt with only by SyzOfGens, we still have the code for shifting components).  
Do we really need syzygies of a list L a 0 element?  
Probably yes, because L might come as a result from another computation...

#### #6 - 09 Feb 2024 08:51 - Anna Maria Bigatti

- Subject changed from SyzOfGens: which shifts for zero generators? to syz, SyzOfGens: which shifts for zero?
- % Done changed from 10 to 30

This is my suggestion: implement a new function syz0 which allows 0s in the input (giving them conventional weights), and leave syz as is, not allowing 0s (and no surprises).

Example:

```
/**/ syz([x, 0, y]); --> error
/**/ syz0([x, 0, y]); --> [0, 1, 0], [y, 0, -x]
```

#### #7 - 09 Feb 2024 10:03 - Anna Maria Bigatti

Should we make SyzOfGens obsolescent? does it do anything more than calling syz(gens(I))? (apart from fixing the 0s)

#### #8 - 09 Feb 2024 10:29 - John Abbott

OK to require that input to **syz** are all non-zero.  
Also OK for a new fn which accepts zero -- not sure how this solves the degree issue mentioned at the start.  
The name **syz0** is OK, but can we find a better/more meaningful name?

**Anna** will check whether SyzOfGens stores a GBasis...

#### #9 - 22 Mar 2024 09:40 - Anna Maria Bigatti

- Related to Design #1649: Add file SparsePolyOps-vector.C added

#### #10 - 26 Mar 2024 09:29 - Anna Maria Bigatti

- Related to deleted (Design #1649: Add file SparsePolyOps-vector.C)

#### #11 - 26 Mar 2024 09:59 - Anna Maria Bigatti

- Related to Feature #1072: syz: apply to ModuleElem? added

#### #12 - 26 Mar 2024 10:42 - Anna Maria Bigatti

Anna Maria Bigatti wrote:

This is my suggestion: implement a new function syz0 which allows 0s in the input (giving them conventional weights), and leave syz as is, not allowing 0s (and no surprises).

I update my suggestion (along the lines of [#1205-2](#)), where F is a suitable FreeModule:  
syz(L) doesn't allow 0 entries  
syz(F, L) allows 0 entries

**#13 - 26 Mar 2024 13:33 - Anna Maria Bigatti**

- *Related to Feature #1805: Add CoCoA-5 syntax: syz(<F,> list of ModuleElem) added*

**#14 - 27 Mar 2024 17:43 - Anna Maria Bigatti**

Anna Maria Bigatti wrote:

Should we make SyzOfGens obsolescent? does it do anything more than calling syz(gens(l))?

I changed my mind. Even though in CoCoALib they are equivalent (gens(l) is a reference), in CoCoA5 it seems pointless to create a CoCoA-5 LIST to be passed to syz.

Moreover, as we do not yet have syz(LIST of MODULEELEM), that's the only way to call it.

So, along the lines that we want CoCoALib and CoCoA-5 to be as similar as possible, I'm now in favour of keeping SysOfGens.

**#15 - 27 Mar 2024 17:47 - Anna Maria Bigatti**

- *Description updated*

- *Status changed from In Progress to Resolved*

- *% Done changed from 30 to 70*

Still to do: documentations showing how to deal with 0 entries, in CoCoALib and in CoCoA5

**#16 - 28 Mar 2024 07:40 - Anna Maria Bigatti**

- *Related to Feature #598: Syzygy for modules: non-homogeneous module added*

**#17 - 28 Mar 2024 08:12 - Anna Maria Bigatti**

- *Description updated*

added documentation for CoCoA-5

**#18 - 28 Mar 2024 08:15 - Anna Maria Bigatti**

- *Description updated*

restored "final decisions" in description removed by mistake.

**#19 - 28 Mar 2024 08:57 - Anna Maria Bigatti**

- *Related to Feature #1809: Make ex-syz.C added*

**#20 - 28 Mar 2024 10:15 - Anna Maria Bigatti**

- *Status changed from Resolved to Closed*

- *% Done changed from 70 to 100*

Updated documentation submodule.txt (need proper example ex-syz.C)

Updated CoCoA-5 test test-syz, cvs-ed and included in official CoCoA-5 tests

Updated CoCoA-5 test test-ArrAndPosets (which uses syz for modules), cvs-ed and included in official CoCoA-5 tests