

CoCoALib - Bug #1379

Fails to recognize zero-dim ideal

27 Dec 2019 12:29 - John Abbott

Status:	Closed	Start date:	27 Dec 2019
Priority:	High	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	100%
Category:	Maths Bugs	Estimated time:	5.55 hours
Target version:	CoCoALib-0.99700	Spent time:	5.55 hours
Description			
Robbiano reports (via email) that CoCoA incorrectly produces "error NYI" with the following input			
<pre>Use P ::= QQ[x,y,z], Lex; I1:= ideal(x-z^2, y-z^3-3*z-1, z^3-z-1); I2:= ideal(x-5, y, z)^2; I3:= ideal(x^2, y-2, z-1); L:= [I1,I2,I3]; I:=IntersectionList(L); dim(P/I); ---> 0 PD:=PrimaryDecomposition(I); --> error NYI</pre>			
Related issues:			
Related to CoCoALib - Bug #1380: Make IsZeroDim more robust		Closed	27 Dec 2019
Related to CoCoALib - Feature #1212: New function: GBasisByHomog		Closed	05 Aug 2018

History

#1 - 27 Dec 2019 12:36 - John Abbott

- Status changed from New to In Progress
- % Done changed from 0 to 10

JAA and Anna have confirmed the bug (with current CVS version).

The problem seems to be that the ideal is not recognized as 0-dim (see line 476 in SparsePolyOps-ideal.C, in function myTestIsPrimary).

JAA notes that the possibly dodgy const_cast is used in several places, including one where the comment is **Tappullus Horribilis** 8-O

JAA reports that the bug persists even when CoCoA is compiled in debug mode.

#2 - 27 Dec 2019 12:38 - John Abbott

- What is the comment on line 528 about?
- What happens if the GB is not a RGB? 8-O
- Or is that impossible?

#3 - 27 Dec 2019 12:47 - John Abbott

I have put a print command in the loop inside lamZeroDim around line 530 in SparsePolyOps-ideal.C.

It reports the following LPPs for some ideal:

```
IamZeroDim START
LPP = y
LPP = x
```

```
LPP = x*z
LPP = x^2
LPP = z^3
IamZeroDim END
```

So the GB is not minimal (nor RGB)!
Clue city!

On my computer the troublesome call to lamZeroDim is the 26th time it is called 8-|

#4 - 27 Dec 2019 13:01 - John Abbott

With GDB, the problem seems to be from a call to lsPrimary on line 735 of SparsePolyOps-ideal-ZeroDim.C inside fn myPrimaryDecomposition_0dim.

I note that the GB of Q[i] was computed using GBasisByHomog; could the result be not a minGB?
Presumably the result of GBasisByHomog is stored in Q[i], and then this same basis is used inside lamZeroDim...

Assuming this is the problem... I see three ways to correct it:

- **(A)** change defn of lamZeroDim so that it works with a non-minGB
- **(B)** change GBasisByHomog so that the result is a minGB (or RGB)
- **(C)** change defn of myPrimaryDecomposition_0dim so that it minimizes the GB of Q[i]

My current preference lies with **(B)**.

Perhaps **(A)** lamZeroDim should also be changed too, since we don't know how the GB was obtained.

#5 - 27 Dec 2019 13:13 - John Abbott

Here is an example where GBasisByHomog returns a non-minGB

```
Use QQ[x,y,z], Lex;
I := ideal(y*z -2*z^5 -2*z^4 +2*z^3, y^2 -16*z^5 -12*z^4 +12*z^3 +12*z^2, x*y -2*x*z -5*y +6*z^5 +2*z^3 -8*z
^2 +10*z, x^2 +10*x*z -10*x -51*z^5 +74*z^4 +16*z^3 -14*z^2 -50*z +25, x*z^2 -x*z +5*z^5 -6*z^4 -4*z^3 +5*z,
z^6 -z^5 -z^4 +z^2, z^3 -z -1);
GBasisByHomog(I);
--> [y -4*z -2, x -z^2, x*z -z -1, x^2 -z^2 -z, z^3 -z -1]
```

NOTE we must update the exbugs in the CoCoA tests after this has been resolved!

#6 - 27 Dec 2019 13:19 - Anna Maria Bigatti

- % Done changed from 10 to 20

John Abbott wrote:

I have put a print command in the loop inside lamZeroDim around line 530 in SparsePolyOps-ideal.C.

It reports the following LPPs for some ideal:

[...]

So the GB is not minimal (nor RGB)!

Clue city!

Indeed!!!

the way lsZeroDim is written requires that the GBasis is minimal (not necessarily RGB), because it just counts the **number** of simple-powers. So, if their number is not equal to **NumIndets**, it returns false, as in this case.

I'm **really surprised** that the GBasis is not minimal, because it's a very simple test in the actual algorithm. That's my next chase.

#7 - 27 Dec 2019 13:22 - Anna Maria Bigatti

John Abbott wrote:

Here is an example where GBasisByHomog returns a non-minGB

[...]

ahhhh!! clue city!!

that might happen with GBasisByHomog!!!

I need to add a minimalization loop after the dehomogenization.

#8 - 27 Dec 2019 14:41 - John Abbott

I'd like the lamZeroDim impl to be more robust; *i.e.* it should work with any GB not just a minGB.

#9 - 27 Dec 2019 15:29 - John Abbott

- Related to Bug #1380: Make lsZeroDim more robust added

#10 - 27 Dec 2019 16:44 - Anna Maria Bigatti

- Related to Feature #1212: New function: GBasisByHomog added

#11 - 28 Dec 2019 18:57 - John Abbott

- Status changed from In Progress to Feedback

- Assignee set to Anna Maria Bigatti

- % Done changed from 20 to 90

- Estimated time set to 5.55 h

Anna revised the code for **GBasisByHomog** (so that the resulting basis is interreduced).

Anna made a new version of CoCoA-5 for Robbiano; Robbiano reports that it seems OK.

#12 - 28 Dec 2019 19:03 - John Abbott

I have added two new tests to **CoCoA-5/tests/exbugs.cocoa5** for this issue.

#13 - 10 Jan 2020 12:17 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

Should there also be more tests where the result is false? Probably little point. I could add a couple of really easy cases...