# CoCoALib - Design #1255

# Ideals with trivial GBasis

11 Mar 2019 19:14 - Anna Maria Bigatti

Status:	New	Start date:	11 Mar 2019	
Priority:	Normal	Due date:		
Assignee:		% Done:	0%	
Category:	Improving	Estimated time:	0.00 hour	
Target version:	CoCoALib-0.99880	Spent time:	0.10 hour	
Description				
If an ideal has a single (non-zero) generator then that generator is automatically a Gbasis. CoCoALib does not currently recognize this. ( <u>#1249</u> ) There are other cases like this: should they be detected?				
Related issues:				
Related to CoCoALib - Feature #1249: principal ideal has a Gbasis			Closed	01 Mar 2019
Related to CoCoALib - Bug #1416: IdealOfProjectivePoints and MinGens			Closed	14 Feb 2020
Related to CoCoALib - Design #1422: Remove flag IhaveGBasisValue?			In Progress	25 Feb 2020
Related to CoCoALib - Feature #1349: ideal ctor where given gens are a gbasis			In Progress	24 Oct 2019
Related to CoCoALib - Design #1647: Suppress zero from ideal generators? Det			Closed	20 Jan 2022

#### History

\_\_\_\_\_

### #1 - 11 Mar 2019 19:16 - Anna Maria Bigatti

[by John Abbott: moved here from #1249]

Also if the gens happen to have coprime LTs wrt to current ordering then they are a GBasis. Might be useful to have a function which checks if the gens are "obviously" a GBasis (without computing anything)?

Also it could be worth computing a GBasis with a low timeout...

Here are some more minor points (after speaking to Anna on the phone):

- Anna was concerned about potential cost if there are many gens;
- if there are more (non-zero) gens than indets then the LTs cannot be coprime;
- when scanning through the list of LTs, if the number of unseen indets is less than the number of remaining (non-zero) gens then they cannot all be pairwise coprime.

## #2 - 11 Mar 2019 19:17 - Anna Maria Bigatti

- Related to Feature #1249: principal ideal has a Gbasis added

#### #3 - 04 Apr 2019 15:00 - John Abbott

Generators which are monomials are also a special case.

Then one could imagine a more complicated "mixed" situation: monomial gens in the Xs, and gens with coprime LTs in Ys...

It seems that the simplest general solution would be to try to compute a GBasis with a low timeout. If we have a proper Buchberger "mill" then every time we call "ComputeGBasis" with a timeout should advance the computation by at least one step. This could lead to a strange situation where repeatedly calling IHaveGBasis(J) produces initially a string of false before suddenly changing to true. Is this desirable? (JAA: probably)

## #4 - 10 Jan 2020 12:31 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

# #5 - 21 Feb 2020 14:53 - John Abbott

- Related to Bug #1416: IdealOfProjectivePoints and MinGens added

## #6 - 25 Feb 2020 17:50 - John Abbott

- Related to Design #1422: Remove flag IhaveGBasisValue? added

# #7 - 28 Oct 2020 23:15 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

# #8 - 29 Oct 2020 12:53 - John Abbott

- Related to Feature #1349: ideal ctor where given gens are a gbasis added

# #9 - 08 Mar 2023 19:49 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880

# #10 - 22 Jan 2024 20:22 - John Abbott

- Related to Design #1647: Suppress zero from ideal generators? Detect 1 and simplify generators? added