

CoCoA-5 - Slug #405

ReducedGBasis not memorized in an ideal

09 Oct 2013 18:17 - John Abbott

Status:	Closed	Start date:	09 Oct 2013
Priority:	Normal	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	100%
Category:	Incomplete function	Estimated time:	2.01 hours
Target version:	CoCoA-5.2.0 spring 2017	Spent time:	2.10 hours
Description			
Sometimes C5 seems not to memorize a computed GBasis -- or is it simply re-interreducing? Anyway, this is an ugly slug!			
Use QQ[a, b, x, y, z], Lex; J:=Ideal(x- a^3 - b^3 - 3, y -a^5 - b^5 - 5, z - a^7 - b^7 - 7); T := CpuTime(); GB_Lex:=ReducedGBasis(J); TimeFrom(T); --> seems to RECOMPUTE the basis: T := CpuTime(); GB_Lex:=ReducedGBasis(J); TimeFrom(T);			
Related issues:			
Related to CoCoALib - Bug #114: Ideals of polynomials		In Progress	26 Mar 2012
Related to CoCoA-5 - Support #977: "universal denominator" (related with Groe...		In Progress	17 Nov 2016
Related to CoCoALib - Feature #957: New function: HasGBasis		Closed	27 Oct 2016
Related to CoCoALib - Feature #961: New function: ReducedGBasis		Closed	03 Nov 2016

History

#1 - 11 Oct 2013 12:52 - Anna Maria Bigatti

- Category set to Incomplete function
- Status changed from New to In Progress
- Target version set to CoCoA-5.1.0 Easter14
- % Done changed from 0 to 10

This is what happens

```
/**/ T := CpuTime(); RGB:=ReducedGBasis(J); DecimalStr(CpuTime()-T);  
69.847  
/**/ T := CpuTime(); RGB:=ReducedGBasis(J); DecimalStr(CpuTime()-T);  
20.480  
/**/ T := CpuTime(); GB:=GBasis(J); DecimalStr(CpuTime()-T);  
0.044
```

The **GBasis** is memorized, but it is not reduced. In fact it is almost reduced:

```
/**/ T := CpuTime(); EqSet([monic(G)|G In GB], RGB); DecimalStr(CpuTime()-T);  
true  
11.465
```

GB has coefficients with denominator 1.
GB and RGB have 119 polynomials, 20 of which with >1000 terms.

Do we want to memorize the reduced GBasis? Or should we have a flag for "tail-reduced"/"monic"?

#2 - 11 Oct 2013 13:01 - Anna Maria Bigatti

ReducedGBasis is defined in package: no wonder it is slow!
I will try to implement it in CoCoALib and see how heavy the reduction actually is.

#3 - 23 Jan 2014 11:39 - John Abbott

Will it be possible to have this fixed by Easter? (*i.e.* before I need it for the Alg Comp course)

#4 - 09 Apr 2014 17:41 - John Abbott

- Target version changed from CoCoA-5.1.0 Easter14 to CoCoA-5.1.1 Seoul14

#5 - 02 Sep 2014 11:37 - John Abbott

- Target version changed from CoCoA-5.1.1 Seoul14 to CoCoA-5.?.?

#6 - 24 Nov 2016 13:05 - John Abbott

Anna now reports that GBasis actually produces an irredundant fully reduced GBasis; the only step missing for making it a reduced GBasis is scaling the polynomials so that they are monic.

#7 - 24 Nov 2016 13:05 - John Abbott

- Related to Support #977: "universal denominator" (related with GroebnerFanIdeals) added

#8 - 24 Nov 2016 13:13 - John Abbott

[translated from an email written in italian]

Here is a potentially important point. For simplicity I consider the case of coeffs in QQ.
So that polynomial reduction can be computed quickly, the reducers should be primitive with integer coeffs. Similarly the polynomial to be reduced should be rescaled to become primitive with integer coeffs; at the end, the final result of the reduction should be rescaled to produce the correct "public" result. If I recall correctly, the code already does this (or most of it).

This means that the elements of the publicly visible RGB will probably be "monic copies" of reducers which are actually used during the reduction. This implies using more memory space -- probably not important in most cases.

An argument analogous to that above presumably applies whenever the coeffs are in a fraction field *e.g.* QQ(a,b)[x,y,z]

#9 - 24 Nov 2016 13:43 - Anna Maria Bigatti

John Abbott wrote:

[translated from an email written in italian]

Here is a potentially important point. For simplicity I consider the case of coeffs in QQ.
If I recall correctly, the code already does this (or most of it).

This is what it does for the internal computation (for all Fraction fields of GCDDomain)

#10 - 24 Nov 2016 13:46 - Anna Maria Bigatti

- Status changed from *In Progress* to *Feedback*
- Assignee set to *Anna Maria Bigatti*
- Target version changed from *CoCoA-5.?.?* to *CoCoA-5.2.0 spring 2017*
- % Done changed from *10* to *90*

GBasis is now exactly ReducedGBasis.

(still an error on RingWeyl, though, where it is not interreduced)

#11 - 12 Jan 2017 15:13 - Anna Maria Bigatti

- Related to Feature #957: New function: *HasGBasis* added

#12 - 27 Apr 2017 14:56 - Anna Maria Bigatti

- Related to Feature #961: New function: *ReducedGBasis* added

#13 - 27 Apr 2017 15:33 - John Abbott

- Description updated

#14 - 27 Apr 2017 15:35 - John Abbott

- Status changed from *Feedback* to *Closed*
- % Done changed from *90* to *100*

Closing after 5 months in feedback. A quick test shows that the RGB is now correctly memorized: well, the example given in the description was instant for the second part.

#15 - 28 Apr 2017 09:14 - Anna Maria Bigatti

- Estimated time set to *2.01 h*