CoCoALib - Slug #1394

Oddly slow GBasis computation (slow final cleanup)

15 Jan 2020 22:51 - John Abbott

Status:	Resolved	**Start date:**	15 Jan 2020
Priority:	Low	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	70%
Category:	Improving	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	1.45 hour

Description

The following computation seems to spend more time in "clean up" than elsewhere; is this reasonable?

```
use P ::= QQ[x,y],Lex;

StartTime := CpuTime();
I := ideal(subst([x^6 -3*x^3*y^2 +x^4,  y^6 +4*x*y^3],y,y+x));
SetVerbosityLevel(110);
RGB := ReducedGBasis(I);
println "RGB time: ", TimeFrom(StartTime);
```

Related issues:

Related to CoCoALib - Slug #777: SLUG: elimination	**In Progress**	**15 Sep 2015**
Related to CoCoALib - Slug #1796: myFinalizeGBasis ("Final clean up") should ...	**New**	**18 Mar 2024**

History

#1 - 07 Feb 2020 12:19 - John Abbott

Here is another example (with **DegRevLex**). The "Final clean up" consumes 99% of the time!

```
-- CoCoA code to generate examples of (0-dim) ideals which contain
-- the binomial x^q-y^p  (if p & q are not coprime, get lower deg binomials).

SetVerbosityLevel(110);
use QQ[x,y,z];
//use ZZ/(32003)[x,y,z];

p := 2;   //  positive integers...
q := 301; //  ...maybe better if they are coprime
f := x^3+x-1;  // f probably should be square-free & deg >= 2, better deg > 2?
//f := x^10+x^9-x^7-x^6-x^5-x^4-x^3+x+1; // Lehmer's polynomial

I1 := ideal(f);
mp := MinPolyQuot(x^p,I1,x);
mq := subst(MinPolyQuot(x^q,I1,x),x,y);
I2 := ideal(mp, mq);
//RationalSolve(gens(I2));

t0 := CpuTime();
I3 := I2 : ideal(x^q-y^p);
println "Colon time: ", TimeFrom(t0);
G := [g in gens(I3) | not(g isin I2)];
NZ := z*product(G)-1;

I4 := I2+ ideal(NZ);
t0 := CpuTime();
x^q-y^p isin I4;
println "isin time: ", TimeFrom(t0);
RGB := ReducedGBasis(I4);
indent([support(g) | g in RGB]); // this seems to be indep of p & q

// General idea:
//   let alpha_1, ..., alpha_d be the roots of f
//   zero set of I2 is ((alpha_j)^p, (alpha_k)^q)  for all j,k
//   poly product(G) is zero when j<>k  in ZeroSet(I2)
```

```
//   ZeroSet(I4)  is  ((alpha_j)^p, (alpha_j)^q)   for all j
//   Hence x^q-y^p is in I4.
```

NOTE if the coeff field is ZZ/(32003) then the whole computation takes 0.1s instead of 300s

#2 - 15 Apr 2021 10:25 - John Abbott

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

#3 - 15 Nov 2021 20:06 - John Abbott

Here is another example using lex:

```
/**/ use QQ[x,y,z,t],lex;
/**/ I := ideal(x^2-7,y^3-11, z^5-13, t-(x+y+z));
SetVerbosityLevel(100);
/**/ GB := ReducedGBasis(I);
```

#4 - 16 Feb 2024 17:30 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99900

#5 - 16 Feb 2024 17:30 - Anna Maria Bigatti

- Target version changed from CoCoALib-0.99900 to CoCoALib-0.99850

Try

```
RGB := GBasisByHomog(I);
```

;-)

#6 - 16 Feb 2024 18:03 - John Abbott

- Related to Slug #777: SLUG: elimination added

#7 - 08 Mar 2024 17:50 - John Abbott

AnnA will look at this!

#8 - 08 Mar 2024 17:53 - Anna Maria Bigatti

- Subject changed from Oddly slow LEX GBasis computation to Oddly slow GBasis computation (slow final cleanup)

- Assignee set to Anna Maria Bigatti

#9 - 16 Mar 2024 21:26 - John Abbott

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

#10 - 18 Mar 2024 18:37 - John Abbott

- Related to Slug #1796: myFinalizeGBasis ("Final clean up") should be more flexible added

#11 - 22 Mar 2024 16:16 - Anna Maria Bigatti

- Description updated

I changed the verbosity to 130 (to see what goes on in the -Final clean up)
Now it is faster because the polynomials going to 0 are detected just by checking the LT (see #777-12).

#12 - 22 Mar 2024 16:43 - Anna Maria Bigatti

- % Done changed from 0 to 40

Current timings after the LT checking in the final cleaning (just marginally better than before, but most of the time seems in the core, not in the final cleaning)

```
use P ::= QQ[x,y],Lex;
StartTime := CpuTime();
I := ideal(subst([x^6 -3*x^3*y^2 +x^4,  y^6 +4*x*y^3],y,y+x));
RGB := GBasis(I);
println "RGB time: ", TimeFrom(StartTime);

RGB time: 56.612

/**/ t0 := CpuTime();
/**/ use QQ[x,y,z,t],lex;
/**/ I := ideal(x^2-7,y^3-11, z^5-13, t-(x+y+z));
/**/ GB := GBasis(I);
/**/ TimeFrom(t0);
18.953
```

JOHN'S TIMES: 57.2 and 20.3 (before checking out Anna's improvements)

#13 - 22 Mar 2024 17:39 - Anna Maria Bigatti

- Description updated

#14 - 22 Mar 2024 17:53 - John Abbott

- Status changed from New to In Progress

Anna points out that the actual cost is the normal form reduction inside the **isin** operator. The computation of the GB is not that slow (less than 0.01s on my linux laptop).

BIG SURPRISE by mistake I re-ran the computation but inside the ring **QQ[x,y,z],lex**, and the **isin** computation took just 1.5s. Odd, because lex is usually much slower!

#15 - 22 Mar 2024 17:56 - Anna Maria Bigatti

- Category changed from Improving to Renaming

- Target version deleted (CoCoALib-0.99880)

John Abbott wrote:

> Anna points out that the actual cost is the normal form reduction inside the **isin** operator. The computation of the GB is not that slow (less than 0.01s on my linux laptop).

Indeed, instead of ~300 s for NF(x^301-y^p, I4);, we have

```
/**/ t0:=CpuTime(); NF(NF(x^25,I4)^12*x-y^p, I4); TimeFrom(t0);
0.125
/**/ t0:=CpuTime(); NF(NF(x^100,I4)^3*x-y^p, I4); TimeFrom(t0);
7.103
```

I cannot see how we can make this fully automatic...

#16 - 22 Mar 2024 18:01 - Anna Maria Bigatti

- Category changed from Renaming to Improving

- Target version set to CoCoALib-0.99880

- % Done changed from 40 to 70

#17 - 22 Mar 2024 18:01 - Anna Maria Bigatti

- Status changed from In Progress to Resolved

#18 - 16 Apr 2024 22:46 - John Abbott

This might be another example where "final clean up" is not instant:

```
use QQ[x,y,z],lex;
//use ZZ/(29641)[x,y,z],lex;
L := [x*y^3-2*y*z-z^2+13,  y^2-x^2*z +x*z^2+3, z^2*x-y^2*x^2+x*y+y^3+12];
I := ideal(L);
SetVerbosityLevel(100);
RGB := ReducedGBasis(I);
```