CoCoALib - Bug #1088

MinPolyQuot: runs out of primes

05 Jul 2017 12:49 - John Abbott

Status:	Closed	**Start date:**	05 Jul 2017
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	Improving	**Estimated time:**	2.00 hours
Target version:	CoCoALib-0.99560	**Spent time:**	3.20 hours

Description

I tried MinPolyQuot on a slightly big example, and it ran out of primes.

This is surely a problem with all modular methods; we must find a good solution.

Related issues:

Related to CoCoA-5 - Feature #1084: New function: PrevPrime	**Closed**	**29 Jun 2017**

History

#1 - 05 Jul 2017 12:51 - John Abbott

Here is an example input which triggers the problem (after about 12 mins' CPU time)

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

DEG := 6;
S := support((1+sum(indets(P)))^DEG);

define RndPoly(S)
  return sum([random(-99,99)*t | t in S]);
enddefine; -- RndPoly

L := [RndPoly(S) | i in 1..3];

I := ideal(L);
IsZeroDim(I);
QB := QuotientBasis(I);
f := RndPoly(QB);
StartTime := CpuTime();
mu := MinPolyQuot(f,I,x);
EndTime := CpuTime();
println "MinPoly time: ", DecimalStr(EndTime-StartTime);
```

#2 - 05 Jul 2017 13:00 - John Abbott

I have just tried running the example from the comment above but with verbosity level 80.
It prints out each prime used and also a polynomial from MinPolyQuotDef.
It would be useful to have the poly print out only at a slightly higher level of verbosity.

#3 - 07 Jul 2017 11:57 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

- % Done changed from 0 to 50

- Estimated time set to 2.00 h

Fixed using PrevPrime instead of NextPrime.
cvs-ed

#4 - 07 Jul 2017 11:58 - Anna Maria Bigatti

- Related to Feature #1084: New function: PrevPrime added

#5 - 07 Jul 2017 11:58 - Anna Maria Bigatti

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99560

#6 - 07 Jul 2017 12:23 - Anna Maria Bigatti

John Abbott wrote:

> I have just tried running the example from the comment above but with verbosity level 80.
> It prints out each prime used and also a polynomial from MinPolyQuotDef.
> It would be useful to have the poly print out only at a slightly higher level of verbosity.

raised to level 85

#7 - 14 Jul 2017 17:04 - John Abbott

- Status changed from New to In Progress

The problem is not yet fixed (but I did have to try "hard" to make a big enough example).

The code should switch from "fast" finite fields to "slow" ones in case of need.
I suppose I must look into it. :-/

#8 - 14 Jul 2017 18:01 - Anna Maria Bigatti

John Abbott wrote:

> The problem is not yet fixed (but I did have to try "hard" to make a big enough example).

??!?!?
all primes from 46000 down to 2 were not enough!?!?!
What monster example have you done?

> The code should switch from "fast" finite fields to "slow" ones in case of need.
> I suppose I must look into it. :-/

That would make lots of code very very tedious....
Have you tried using "MinPolyDefQuot" (which is not modular)?

#9 - 14 Jul 2017 21:28 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 50 to 80

I think I have fixed the code: the only real change needed was to use NewZZmod instead of NewRingFp.
This allows the use of primes above 46000 (or whatever the previous limit was). For small examples there is no measurable speed penalty.

Note that NewZZmod tries to be clever: it will actually call NewRingFp if possible.
I have made it clearer in the documentation for RingFp that one should normally use NewZZmod.

I'll let Anna check the code quickly, and move it to "feedback" if she is convinced.

#10 - 08 Nov 2017 14:39 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100