

CoCoALib - Support #438

Polynomial multiplication (product of RingElem)

08 Feb 2014 15:33 - John Abbott

Status:	In Progress	Start date:	08 Feb 2014
Priority:	Normal	Due date:	
Assignee:		% Done:	10%
Category:	Various	Estimated time:	0.00 hour
Target version:	CoCoALib-1.0	Spent time:	0.50 hour
Description			
Bruns asks whether $f*g$ and $g*f$ take the same time or whether one way round is better than the other. He mentions the special case of one polynomial actually being a monomial; it probably is worth handling this case specially in the code.			
Related issues:			
Related to CoCoA-5 - Support #242: CoCoA-5 Projects for students (e.g. credit...		In Progress	28 Sep 2012

History

#1 - 08 Feb 2014 15:40 - John Abbott

[from an email sent to Bruns] I do recall that there are complications for inserting the terms in the correct order; a naive implementation has cubic complexity in some cases. One tricky case is $f := (x^N-1)/(x-1)$ and $g := \text{subst}(f,x,x^N)$; where N is an integer parameter. Both f and g have N terms, while the product contains N^2 terms. CoCoALib's use of geobuckets should help considerably in any case.

#2 - 09 Feb 2014 09:50 - Anna Maria Bigatti

- Status changed from New to In Progress
- Target version set to CoCoALib-0.99534 Seoul14
- % Done changed from 0 to 10

In SparsePolyRing.C -- myMul there is this code

```
if (IamCommutative() && myNumTerms(rawf) > gLen)
{ myMul(rawlhs, rawg, rawf); return; }
```

Before it it checks whether one of the two is constant.
I can add the check in case one is a single summand.

#3 - 09 Feb 2014 12:48 - Winfried Bruns

I think it is a question that needs careful testing before any changes should be done. As observed by John and mentioned in my last mail, the use of the geobucket presumably levels out the differences between F^*G and G^*F .

#4 - 09 Feb 2014 16:08 - Anna Maria Bigatti

Winfried Bruns wrote:

I think it is a question that needs careful testing before any changes should be done. As observed by John and mentioned in my last mail, the use of the geobucket presumably levels out the differences between F^*G and G^*F .

What I meant is that all the work we've been talking about has already been done (2008):

- check if one poly is a constant (and mult is commutative)
- check if one poly is longer (and mult is commutative)
- check if one poly is a single summand (and mult is commutative)
- use geobucket

Only optimization to be done: reorganize code so that myNumTerms is computed fewer times.

#5 - 09 Feb 2014 16:52 - Winfried Bruns

I am sorry for having raised a non-issue.

Regarding NumTerms: is it a data field of the polynomial or is it computed when asked for?

#6 - 10 Jul 2014 16:28 - John Abbott

- Target version changed from *CoCoALib-0.99534 Seoul14* to *CoCoALib-1.0*