

CoCoALib - Slug #884

DistrMPolyInIPP::myPushFront and DistrMPolyInIPP::myPushBack inefficient if arg is a PP

24 May 2016 15:20 - John Abbott

Status:	New	Start date:	24 May 2016
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Improving	Estimated time:	0.00 hour
Target version:	CoCoALib-1.0	Spent time:	0.50 hour
Description			
In the class DistrMPolyInIPP the mem fns myPushFront and myPushBack@ can accept raw coeff and raw pp, BUT the impl then "explodes" the PP into a vector of exponents then calls the myPushFront or myPushBack mem fns on the vector. This is rather inefficient!			
Find a way to copy the already encoded PP!			
Related issues:			
Related to CoCoALib - Slug #881: ReadExpr is too slow on large polys		Closed	09 May 2016

History

#1 - 24 May 2016 15:25 - John Abbott

I wrote a special reading fn for polynomials (using a "private" linearization). It was not as fast as I hoped, and profiling showed that a lot of time was spent in DistrMPolyInIPP::myPushFront.

Looking at the code I saw that it "explodes" the PP into an expv, and then recompresses it into an ordv. This is a perceptible waste of time!

The source code for myPushFront contained a commented-out line calling a straight assignment; activating the line triggered a compilation error. The problem is that the PP which is passed in is of type PPMonoidElemConstRawPtr but the assignment wants a const OrdvElem*. How to do what I want reasonably cleanly and safely?

#2 - 24 May 2016 15:25 - John Abbott

- Related to Slug #881: ReadExpr is too slow on large polys added