

CoCoALib - Feature #1022

New "LF" function which is based on StdDeg

06 Mar 2017 16:44 - John Abbott

Status:	New	Start date:	06 Mar 2017
Priority:	Normal	Due date:	
Assignee:		% Done:	20%
Category:	New Function	Estimated time:	3.00 hours
Target version:	CoCoALib-1.0	Spent time:	1.20 hour
Description			
[2017-03-06] Bruns requested by email to have a function like LF which works with respect to the standard grading (even in an ungraded ring, such as one with lex ordering).			
Related issues:			
Related to CoCoALib - Slug #1042: LF curiously slow (breaking a poly into hom...		Closed	10 Apr 2017
Related to CoCoA-5 - Feature #1439: New function: LinearForm		Closed	07 Mar 2020

History

#1 - 06 Mar 2017 16:50 - John Abbott

CoCoALib currently offers `deg` (or equiv. `StdDeg`) and `wdeg`. These functions throw an exception if the ring has grading dim zero (*i.e.* ungraded). Bruns would like to have a new function which determines the "leading form with respect to the standard grading" regardless of any grading which the ring has.

He also pointed out that to be consistent we should actually call the current **LF** function **wLF** (akin to `wdeg`).

I noticed that in one of his messages he actually wanted a decomposition of a polynomial into homogeneous pieces; should we consider adding such a function?

#2 - 06 Mar 2017 17:50 - Winfried Bruns

Why not have a function for decomposing a polynomial into its homogeneous components? There is one in `NmzIntegrate`.

#3 - 06 Mar 2017 19:14 - Anna Maria Bigatti

Here is the function.

I'm still uncertain whether this should be part of `cocoalib` or not.

Anyway it may be copied and used :-)

```
RingElem LF_StdDeg(ConstRefRingElem f)
{
  if (!IsPolyRing(owner(f)))
    CoCoA_ERROR(ERR::NotElemPolyRing, "LF_StdDeg(f)");
  const SparsePolyRing P = owner(f);
  if (IsZero(f)) CoCoA_ERROR(ERR::ZeroRingElem, "LF_StdDeg(f)");
  RingElem F(P);
  long MaxDeg = 0;
  for (SparsePolyIter it=BeginIter(f) ; !IsEnded(it) ; ++it )
  {
    long d = deg(PP(it));
    if (d > MaxDeg)
    {
      MaxDeg = d;
      F = monomial(P, coeff(it), PP(it));
    }
    else if (d == MaxDeg)
      F += monomial(P, coeff(it), PP(it));
  }
}
```

```
return F;  
}
```

#4 - 06 Mar 2017 19:37 - Anna Maria Bigatti

- % Done changed from 0 to 20
- Estimated time set to 3.00 h

Winfried Bruns wrote:

Why not have a function for decomposing a polynomial into its homogeneous components? There is one in `NmzIntegrate`.

I'm reluctant about that. We usually follow the rule not to make easy functions for things which might be pointlessly expensive in their application. `deg` is quite expensive in `cocoalib`. moreover it would duplicate piece by piece the whole polynomial (from what I see in you particular example you need only the highest degree part) That's why we have a **SparsePolyIter** which is so cheap and nice to use ;-)

#5 - 10 Apr 2017 11:29 - John Abbott

- Related to Slug #1042: LF curiously slow (breaking a poly into homog pieces) added

#6 - 09 Mar 2020 16:53 - Anna Maria Bigatti

- Related to Feature #1439: New function: LinearForm added