CoCoALib - Feature #1778

Homogenizer

02 Feb 2024 20:04 - John Abbott

Status:	In Progress	**Start date:**	02 Feb 2024
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	10%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	1.20 hour

Description

Here is a suggestion for a cleaner, simpler way to effect homogenization.
The main idea is to create a new type of object: a **homogenizer**.
The result is an object which can be applied as a function to a poly, and will return the homogenization.
It can also be applied to an ideal, and return the homogenization of the ideal.

Of course, the code that does the work will remain essentially the same, but
I am hoping it will be easier to use.

Related issues:

Related to CoCoALib - Slug #1737: Homogenization of an ideal with ZZ^1-grading	**In Progress**	**29 Apr 2023**

History

#1 - 02 Feb 2024 20:10 - John Abbott

I am not yet sure of the details, but I am hoping that it can be used as in this example:

```
ring P = NewPolyRing(...);
RingElem f(P, "x^2-y");
homogenizer H(P); // optional second arg saying where the homog indet should be (first/last)
RingElem f_hom = H(f);
ideal I = ideal([...]);
ideal I_hom = H(I);
```

The ctor for homogenizer will internally create the extended poly ring, P_hom, and
also the mapping of indeterminates of P into P_hom. It will also be possible to
use the homogenizer to dehomogenize (not sure what syntax to use).

#2 - 02 Feb 2024 20:13 - John Abbott

Following KISS, it is probably best to try the simple interface first.
If it works well, and we like it, then we can consider offering more
options: *e.g.* letting the caller specify the extended poly ring, and
how the indeterminates should map.

This way all sanity checking can be done in the ctor for homogenizer,
and all later calls will not need to check much.

#3 - 02 Feb 2024 20:13 - John Abbott

- Related to Slug #1737: Homogenization of an ideal with ZZ^1-grading added

#4 - 02 Feb 2024 20:21 - John Abbott

The ctor for homogenizer will create the extended polynomial ring.
What properties should this extended ring have?
Same coeff ring, of course.
Term ordering? Presumably compatible with that of the starting ring,
and such that the homogenizing indets have weights like the columns in
an identity matrix. But what happens below the weights in the order
matrix for the big poly ring? Maybe it depends on the intended use?

#5 - 08 Feb 2024 22:08 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I have implemented homogenizer in OSCAR, and am happy with the interface it gives. Also some others are using it, and they seem to be happy too.
So it would be good to add somethinh analogous to CoCoALib :-)

#6 - 25 Apr 2024 21:54 - John Abbott

This is for **JOHN** Knuckle down and do it!

#7 - 09 May 2024 20:58 - John Abbott

The interface in OSCAR lets the caller choose where the "new" indeterminates appear in the list of indets of the bigger ring.
In most cases the caller chooses "before all the others" or "after all the others". What do we want to do?