CoCoALib - Design #647

Unique copies of free modules?

10 Nov 2014 16:36 - John Abbott

Status:	New	**Start date:**	10 Nov 2014
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Safety	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	0.75 hour

Description

Discuss having unique copies of free modules; or at least that the default creation mechanism should not produce distinct canonically isomorphic free modules.

Related issues:

Related to CoCoALib - Feature #482: Unique copies of rings -- smart ctor	**In Progress**	**19 Mar 2014**
Related to CoCoA-5 - Design #646: Unique copies of free modules?	**New**	**10 Nov 2014**

History

#1 - 10 Nov 2014 16:45 - John Abbott

After speaking to Anna... here are some aspects to consider:

- a major use of modules in CoCoA is for syzygy modules (which have shifts in the homogeneous case) -- what to do about free modules which are identical except for the shifts?

#2 - 10 Nov 2014 16:51 - John Abbott

Currently I'm considering offer two ways of "creating" a free module:

1. "create" a unique free module (of dim n over ring R) -- a new free module is created the first time, thereafter a reference to that module is returned.
2. "create" a new free module (distinct from all other free modules created so far)

Approach (1) appears to need a sort of global registry; this is not ideal in a multithreaded setting, but I cannot imagine it'd ever be a bottleneck (who would ever create zillions of free modules?)

An alternative to approach (2) would be to have a function which "clones" an existing free module (perhaps changing the shifts?). Not sure this is a good idea.

#3 - 10 Nov 2014 16:55 - John Abbott

Are the concepts of **free module with shifts** and **free module without shifts** distinct?

Note that the notion of shifts makes sense only if the underlying ring has a notion of degree.

If the two concepts are distinct; do we want to be able to "view" a free module with shifts as a free module without them? If so, how? C++ subclass?