CoCoALib - Design #297

Modules design: brainstorming

25 Jan 2013 11:41 - Anna Maria Bigatti

Status:	In Progress	**Start date:**	25 Jan 2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	30%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	4.25 hours

Description

Currently (25th Jan 2013) we have this inheritance:

```
module -> FGModule -> submodule
                   -> FreeModule
                   -> QuotientModule
```

The structure is not yet as rich as for rings/ideals and it is intrinsically more complicated.

This issue is just to collect any idea on modules which should then become properly designed new issues.

Related issues:

Related to CoCoA-5 - Feature #200: add modules and module operations	**Closed**	**29 Jun 2012**
Related to CoCoALib - Feature #304: Module ordering and grading (and shifts)	**Closed**	**11 Feb 2013**
Related to CoCoALib - Design #305: FreeModule: unique copy?	**In Progress**	**12 Feb 2013**
Related to CoCoALib - Bug #335: Equality of (sub)modules	**Closed**	**08 Apr 2013**
Related to CoCoALib - Design #703: Add more operations between modules (CoCoA...	**New**	**15 May 2015**
Related to CoCoALib - Design #871: Redesign ideals	**New**	**26 Apr 2016**

History

#1 - 30 Jan 2013 11:43 - Anna Maria Bigatti

I added (in CoCoA-5) the function **ModuleOf** analogue for **RingOf**
Does it make sense? for a ModuleElem yes, for a submodule yes, ... but in cocoa-5 we do not (yet) distinguish modules from submodules.... and should we want to?

#2 - 30 Jan 2013 11:51 - Anna Maria Bigatti

It would make lots of sense a syntax for creating a submodule from a matrix.
I suggest the syntax **SubmoduleCols(F,M)**, **SubmoduleRows(F,M)** (FreeModule and matrix)
I think it should be available in CoCoALib.

Right now we have **NewSubmodule** obeying the rule that calling it twice will return two unequal objects (in the sense of **M1 != M2**), but, as for ideals, we can test equality of submodules of the same free module (at least for modules on polynomials), so: should we allow equality tests? I believe so.

#3 - 30 Jan 2013 18:01 - Anna Maria Bigatti

I think **GensToCols**/**GensToRows** would be quite useful functions, but what would be the actual meaning? It should represents the coordinated wrt a set of generators in the module it is a submodule of: so only for submodules?
or wrt to the free module? (then, what should **gens** do?)

Mathematically modules a bit too flexible and that makes formalizing a bit too delicate...

#4 - 23 May 2013 08:15 - Anna Maria Bigatti

- Status changed from New to In Progress

- % Done changed from 0 to 30

#5 - 31 Jul 2013 15:54 - Anna Maria Bigatti

added **IsHomog(M)**, **IsZero(M)**, **LT** for M (sub)module.
added test for **IsHomog** in **test-GOperations2**

#6 - 01 Apr 2014 17:35 - Anna Maria Bigatti

- Target version set to CoCoALib-0.99533 Easter14

#7 - 07 Apr 2014 18:15 - John Abbott

- Target version changed from CoCoALib-0.99533 Easter14 to CoCoALib-0.99534 Seoul14

#8 - 31 Jul 2014 14:55 - John Abbott

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

#9 - 26 Apr 2016 15:11 - John Abbott

- Related to Design #871: Redesign ideals added