

CoCoALib - Design #1065

Design: should we allow an ideal to change ring?

11 May 2017 08:26 - Anna Maria Bigatti

Status:	Closed	Start date:	11 May 2017
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Data Structures	Estimated time:	2.20 hours
Target version:	CoCoALib-0.99560	Spent time:	2.10 hours
Description I have some trouble writing a bit of code because I cannot redefine an ideal in another ring (changing characteristic). Should we allow that, as we did for RingElem? Probably not...			
Related issues: Related to CoCoALib - Bug #1064: Bug in MinPolyModular (ugly prime) Closed 10 May 2017			

History

#1 - 11 May 2017 08:32 - Anna Maria Bigatti

This would be convenient, but also potentially dangerous:

In the case of RingElem the empty constructor gives 0 in ZZ, and this is *mathematically* (but not automatically) mapped into any other ring.

Similarly the empty ideal constructor should give the ideal 0 in ZZ, but this is not quite the same situation.

Having many rings around is delicate. I'm very undecided.

#2 - 11 May 2017 08:43 - Anna Maria Bigatti

- Related to Bug #1064: Bug in MinPolyModular (ugly prime) added

#3 - 11 May 2017 08:48 - Anna Maria Bigatti

I managed to write my loop changing rings cleanly (modular computation).

While changing characteristic I forcibly had a common ground (in my case the original ring P over QQ), so the check for the loop condition could be in P, keeping the changing ideal within the loop body.

This is indeed cleaner design!!

Moral: I have been forced to think to work around it, and the result was cleaner code.

#4 - 16 Jun 2017 16:48 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The students here working on translating HomomorphismFns.cpkg5 into CoCoALib wanted to reassign an ideal changing ring. The code was something like this:

```
ideal I;  
if (IsPolyRing(R)) { I = ideal(zero(R)); }  
else { /*R is QuotientRing*/ I = DefiningIdeal(R); }
```

The above code causes problems for CoCoALib because currently an ideal variable may contain ideals only from the ring of the initial value given to that variable.

The simple solution (allowing the students to follow the structure of the CoCoA-5 package) would be to allow ideal variables to be assigned any ideal, *i.e.* without any restriction on the ring to which the ideal must belong.

If we maintain the restriction then the code must be reorganized (presumably by splitting the single fn into several functions, one for each subcase).

Personally I do not see any advantage in keeping the restriction on assigning ideals; it seems to make CoCoALib awkward to use without giving any (obvious) benefit.

#5 - 27 Jun 2017 22:29 - Anna Maria Bigatti

John Abbott wrote:

The students here working on translating HomomorphismFns.cpkg5 into CoCoALib wanted to reassign an ideal changing ring. The code was something like this:
[...]

a quick trick I used in similar cases is to make an external function:

```
ideal IdealFromRing(const ring& R)
{
  if (IsPolyRing(R)) return ideal(zero(R));
  /*R is QuotientRing*/ return DefiningIdeal(R);
}
```

Anyway I agree that, after allowing a RingElem to change ring, we can do the same for ideals.

#6 - 28 Jun 2017 13:44 - John Abbott

- % Done changed from 10 to 20

An important point in Anna's previous comment is that her function is a "trick". It seems unreasonable to require the user to write an "ad hoc" global function (presumably in an anonymous namespace?) to achieve something which should be trivial.

#7 - 19 Jul 2017 19:47 - John Abbott

- Status changed from *In Progress* to *Feedback*

- Assignee set to *John Abbott*

- Target version changed from *CoCoALib-1.0* to *CoCoALib-0.99560*

- % Done changed from 20 to 90

I have implemented the change: just delete the check inside operator= for ideals. Some simple tests passed (e.g. the students with their code). So I'm putting this in feedback.

#8 - 08 Nov 2017 14:22 - John Abbott

- Status changed from *Feedback* to *Closed*

- % Done changed from 90 to 100

#9 - 08 Nov 2017 16:29 - John Abbott

- Estimated time set to 2.20 h

#10 - 07 Sep 2023 08:25 - Anna Maria Bigatti

- Description updated