CoCoALib - Feature #645

Automatic mapping of RingElem: user selectable at run-time (GlobalManager?)

04 Nov 2014 11:28 - John Abbott

Status:	Rejected		**Start date:**	04 Nov 2014
Priority:	Normal		**Due date:**	
Assignee:	John Abbott		**% Done:**	100%
Category:	Safety		**Estimated time:**	1.11 hour
Target version:	CoCoALib-0.99850		**Spent time:**	1.10 hour

Description

I suggest that in CoCoALib we give the user the option to have automatic mapping of RingElem values.

For instance, to write a first prototype of a function it would be handy not to have to worry about mapping between coeffring and polyring. Later on, once the code is working, automatic mapping can be disabled (for better speed, or perhaps better cleanliness), and then the code modified so that all mapping is explicit.

This suggestion implies a global flag which has to be tested whenever there is a ring mismatch between arguments. The run-time cost should be negligible, and the extra implementation beyond the code for automatic mapping should be minimal (just an if statement).

What do you think?

Related issues:

Related to CoCoA-5 - Feature #7: Automatic mapping between (some) rings	**Resolved**	**20 Oct 2011**
Related to CoCoALib - Feature #223: Automatic mapping of RingElems	**Closed**	**08 Aug 2012**
Related to CoCoA-5 - Feature #1461: Automatic mapping for multiplication?	**Closed**	**10 Jun 2020**
Related to CoCoA-5 - Design #637: Undesirable consequence of automatic mappin...	**Closed**	**23 Oct 2014**
Related to CoCoA-5 - Design #635: Automatic mapping of RingElem (in operation...	**Closed**	**22 Oct 2014**
Related to CoCoA-5 - Design #1493: Automatic ring mapping in assignment?	**Closed**	**28 Sep 2020**
Related to CoCoALib - Design #1689: Automatic mapping of RHS in += etc	**Closed**	**15 Jun 2022**

History

#1 - 04 Nov 2014 11:34 - John Abbott

Since the flag would have to be global, it makes sense to put it inside the GlobalManager.

If the flag is simply set when the GlobalManager is created, and cannot be changed thereafter, then there should be no problems in multi-threaded environments. That said, I could imagine that someone may want to turn on/off automatic mapping inside a function (but this would not be threadsafe) - though that would be a rather "unclean" approach!

If we want automatic mapping in CoCoA-5, then it simply needs to say so when it creates its internal GlobalManager.

#2 - 04 Nov 2014 17:24 - Anna Maria Bigatti

I'm a bit uneasy about this....
Making the automatic mapping working generally would involve a lot of code.... or are you thinking of just a few operations like "*"? ("*" anyway is a lot of operations thinking of all the combintations)
We could try, but I'm uneasy....

#3 - 04 Nov 2014 17:46 - John Abbott

What I should have said originally is: **if** *we decide to do automatic mapping of RingElem in CoCoALib* **then** *we should consider making a flag to turn it on/off.*

I think that if we want CoCoALib to have wider acceptance then we have no choice but to implement automatic mapping (I guess only from a subring to a superring, not more generally). If we do it right then it should not require that much code: *so simple that there are obviously no defects*...

#4 - 10 Jun 2020 16:07 - Anna Maria Bigatti

- Related to Feature #1461: Automatic mapping for multiplication? added

#5 - 27 Oct 2020 19:17 - John Abbott

- Status changed from New to In Progress

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99850

- % Done changed from 0 to 10

At the moment all fns which allow automatic promotion of ringelems call the function **AutomaticConversionHom**.

It might suffice simply to put a check for the allow/forbid flag inside the function.

To avoid problems in a multithreaded environment, it is better that the flag be set once when creating the GlobalManager (and cannot thereafter be altered).

At the moment I have some doubts as to whether it is a good idea to let the user disable auto promotion. It is definitely handy being able to write things like f/LC(f)...

#6 - 15 Jun 2022 22:22 - John Abbott

Now I am uncertain about this idea.
Code which takes advantage of automatic promotion tends to look neater, and is easier to read.
I very much doubt that there is a significant speed difference.

This evening I'm tempted to suggest **rejecting** this proposal.

#7 - 16 Feb 2024 17:32 - Anna Maria Bigatti

- Related to Design #637: Undesirable consequence of automatic mapping of RingElems? added

#8 - 16 Feb 2024 17:33 - Anna Maria Bigatti

- Related to Design #635: Automatic mapping of RingElem (in operation with a compound value) added

#9 - 16 Feb 2024 17:33 - Anna Maria Bigatti

- Related to Design #1493: Automatic ring mapping in assignment? added

#10 - 16 Feb 2024 17:35 - Anna Maria Bigatti

- Related to Design #1689: Automatic mapping of RHS in += etc added

#11 - 16 Feb 2024 17:41 - John Abbott

- Status changed from In Progress to Rejected

- Assignee set to John Abbott

- % Done changed from 10 to 100

- Estimated time set to 1.11 h

We have auto mapping in many places without any real problems.
If the user disables auto mapping then this may cause problems in many existing functions.
Anna expressed concern about unexpected mapping an element automatically into a ring with undesired properties *e.g.* working in QQ when one expected to be in ZZ.