CoCoALib - Feature #113

Introduce PartialHom

23 Mar 2012 17:02 - John Abbott

Status:	In Progress		Start date:	23 Mar 2012
Priority:	Normal		Due date:	
Assignee:	John Abbott		% Done:	10%
Category:	New Function		Estimated time:	15.00 hours
Target version:	CoCoALib-1.0		Spent time:	10.25 hours

Description

Currently CoCoALib does not distinguish between proper homs and partial homs.

Thus we can create a (partial) RingHom from QQ to ZZ which is valid only for integers in QQ. We can create a RingHom from ZZ to QQ -- a genuine RingHom.
Composing them produces the identity RingHom from QQ to QQ; but this does not have the same behaviour as applying first the partial hom and then the proper hom!

We should introduce separate types for full and partial homs. If this proves to be problematic, a lesser solution would be to add a mem fn saying whether it is full or partial.

Related issues:

Related to CoCoALib - Feature #562: Subrings	**New**	**20 May 2014**
Related to CoCoA-5 - Design #635: Automatic mapping of RingElem (in operation...	**Closed**	**22 Oct 2014**

History

#1 - 31 Aug 2012 14:39 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

JAA has made a first (incomplete) implementation.

Currently the names chosen are RingHom and PartialRingHom. It seems sensible to retain the name RingHom for true homomorphisms. Other ideas for partial ring homs are SubringHom and RingMap. Any suggestions regarding naming are welcome.

The current design is as follows:

- an abstract base class RingHomBase; all implementation classes (incl partial homs) are derived from this class
- added a virtual member fn IamPartial to RingHomBase
- PartialRingHom is a (smart) pointer to an instance of type RingHomBase
- RingHom is derived from PartialRingHom [note that it satisfies the "is-a" relationship] so that the compiler can easily convert from RingHom to PartialRingHom; a RingHom guarantees that its impl gives false as the result a call to IamPartial
- assignment to RingHom from PartialRingHom is allowed but checks at run-time that the impl gives false as the result of a call to IamPartial

This design was chosen because it was the first that came mind. JAA is not entirely happy because the distinction between RingHom and PartialRingHom does seem somewhat artificial.

#2 - 28 Jan 2013 08:10 - Anna Maria Bigatti

- Category set to New Function

#3 - 01 Apr 2014 17:41 - Anna Maria Bigatti

- Target version set to CoCoALib-0.99533 Easter14

#4 - 07 Apr 2014 14:41 - John Abbott

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

- % Done changed from 0 to 10

My impl attempt failed; static typing in C++ did not help.

Now considering defining notion of subring (certain limited families!)

#5 - 09 Jul 2014 18:04 - John Abbott

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