CoCoA-5 - Feature #7

Automatic mapping between (some) rings

20 Oct 2011 11:48 - Anna Maria Bigatti

Status:	Resolved	**Start date:**	20 Oct 2011
Priority:	High	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	80%
Category:	Parser/Interpreter	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.4.2	**Spent time:**	7.00 hours

Description

There are a few obvious cases, i.e. those for which there is a "single step" homomorphism, for example:
RingZ --> R
RingQ --> R
R --> R/I
R --> R[x,y,z]

AMB 2012 added **matrix(R, M)** to CoCoA-5 for automatic mapping
AMB 2013 added **RingElem(R, a)** to CoCoA-5 for automatic mapping
AMB 2014 added **ReadExpr(R, "...")** to CoCoA-5 for reading an explicit expression

Related issues:

Related to CoCoALib - Feature #223: Automatic mapping of RingElems	**Closed**	**08 Aug 2012**
Related to CoCoALib - Feature #151: Iterated CanonicalHom	**New**	**03 May 2012**
Related to CoCoA-5 - Bug #100: BringIn should map only the indets in its arg	**New**	**07 Mar 2012**
Related to CoCoA-5 - Bug #132: BringIn should allow reduction modulo p	**Closed**	**17 Apr 2012**
Related to CoCoA-5 - Feature #309: (Multi)BlockMatrix	**Closed**	**13 Feb 2013**
Related to CoCoA-5 - Design #483: Unique copies of rings in CoCoA-5	**New**	**19 Mar 2014**
Related to CoCoALib - Feature #209: ReadExpr: input polynomials in CoCoALib	**Closed**	**24 Jul 2012**
Related to CoCoA-5 - Design #634: Symbol in the coeff ring	**Rejected**	**22 Oct 2014**
Related to CoCoA-5 - Design #635: Automatic mapping of RingElem (in operation...	**Closed**	**22 Oct 2014**
Related to CoCoA-5 - Design #636: Distinguish indets from symbols in coeffrin...	**Rejected**	**23 Oct 2014**
Related to CoCoA-5 - Design #637: Undesirable consequence of automatic mappin...	**Closed**	**23 Oct 2014**
Related to CoCoALib - Feature #645: Automatic mapping of RingElem: user selec...	**Rejected**	**04 Nov 2014**
Related to CoCoALib - Feature #1132: Canonical homomorphism for (some) polyno...	**New**	**30 Nov 2017**
Related to CoCoALib - Feature #738: Extend homomorphism to polynomial ring	**In Progress**	**25 Jun 2015**
Related to CoCoALib - Design #1085: Fns with "OUT" args: should they give ERR...	**Closed**	**30 Jun 2017**
Related to CoCoA-5 - Feature #1461: Automatic mapping for multiplication?	**Closed**	**10 Jun 2020**
Related to CoCoALib - Design #1414: Make class RingElemVector?	**In Progress**	**12 Feb 2020**
Related to CoCoA-5 - Design #1493: Automatic ring mapping in assignment?	**Closed**	**28 Sep 2020**
Related to CoCoALib - Design #1515: Indets in coeffring are ringelems in coef...	**Rejected**	**22 Oct 2020**

History

#1 - 08 Nov 2011 16:41 - John Abbott

JAA thinks that a good design approach would be to produce a function which takes as args two rings, and returns a **RingHom** (and not a partial ring hom) which must be applied to map elements of the "simpler" ring into the other ring. The domain and codomain of the ringhom will indicate which value(s) is/are to be mapped.

JAA is not sure what the function should do if no automatic mapping is supported between the two rings. There are two obvious alternatives: throw an exception, or return a boolean (in addition to the RingHom).

JAA is not sure what the function should do if the two rings supplied are equal -- it could return the identity RingHom or it could give an error. At the moment giving an error seems the better option.

#2 - 15 Nov 2011 14:00 - Anna Maria Bigatti

- Category set to Parser/Interpreter

#3 - 23 Mar 2012 14:41 - John Abbott

Probably the simplest definition is simply that R1 is "simpler" than R2 if the construction of R2 passed explicitly via that of R1.

For instance this means that R is **always** regarded as being simpler than the quotient R/I.

#4 - 30 May 2012 13:09 - Anna Maria Bigatti

Proposal for new "constructors": R ring, A matrix, x ringelem (or powerproduct or ...), ... and what else?

Mat(R, A), **RingElem(R, x)**
checks if **object** is mappable into **R** and returns its image in **R**: equivalent to

```
phi := CanonicalHom(RingOf(object), R);
apply(phi, object)
```

Automatic ring in operations is in general very difficult to detect.
Making these new constructors available might be a good and clean intermediate solution.

These "constructors" may be made available in CoCoALib too, but that implies including CanonicalHom.H in all **.C** files offering them.

#5 - 30 May 2012 16:08 - John Abbott

It would be handy to have a similar shortcut for homogeneous lists.
Also for ideals, but that raises the same question that came up for **BringIn**.

The doc for BringIn says that it works for ringelems and matrices, lists and vectors but **not** ideals.
The CoCoALib fn **apply** does not appear to be documented!

#6 - 25 Sep 2012 16:14 - John Abbott

- Status changed from New to In Progress

- Assignee set to Anna Maria Bigatti

- % Done changed from 0 to 10

#7 - 04 Oct 2012 12:12 - John Abbott

Assignment of an integer (or rational) to a RingElem will automatically map into the ring.

However, we want to be cautious about extending this automatic mapping to other types of assignment, especially since assignment from a RingElem belonging to another ring does not perform automatic mapping.

#8 - 15 Feb 2013 11:46 - Anna Maria Bigatti

- Target version set to CoCoA-5.0.9

#9 - 05 Dec 2013 11:47 - John Abbott

Renzo reminds us that this task is important!

#10 - 21 Mar 2014 11:48 - Anna Maria Bigatti

- Target version changed from CoCoA-5.0.9 to CoCoA-5.1.0 Easter14

#11 - 09 Apr 2014 17:38 - John Abbott

- Target version changed from CoCoA-5.1.0 Easter14 to CoCoA-5.1.1 Seoul14

#12 - 27 Aug 2014 18:36 - Anna Maria Bigatti

- Target version changed from CoCoA-5.1.1 Seoul14 to CoCoA-5.1.2 summer 2015

- % Done changed from 10 to 30

#13 - 11 May 2015 14:53 - John Abbott

- Target version changed from CoCoA-5.1.2 summer 2015 to CoCoA-5.1.3/4 Jan 2016

#14 - 17 Feb 2016 11:35 - John Abbott

- Target version changed from CoCoA-5.1.3/4 Jan 2016 to CoCoA-5.2.0 spring 2017

I am "bumping" this issue by 1 version of CoCoA-5. It is marked as "high" priority, but has not been worked on for 3-4 years (no doubt because it is tricky).

Whether we'll really be able to deal with it for the next version remains to be seen.

#15 - 13 Oct 2016 17:25 - John Abbott

No chance of finishing this is in the near future; postponing by 1 version.

#16 - 13 Oct 2016 17:25 - John Abbott

- Target version changed from CoCoA-5.2.0 spring 2017 to CoCoA-5.2.2

#17 - 15 Nov 2017 17:41 - John Abbott

- Target version changed from CoCoA-5.2.2 to CoCoA-5.2.4

#18 - 30 Nov 2017 10:00 - John Abbott

- Related to Feature #1132: Canonical homomorphism for (some) polynomial rings? added

#19 - 30 Nov 2017 10:02 - John Abbott

- Related to Feature #738: Extend homomorphism to polynomial ring added

#20 - 25 Jul 2018 17:15 - John Abbott

- Target version changed from CoCoA-5.2.4 to CoCoA-5.3.0

#21 - 01 Oct 2019 14:54 - John Abbott

- Target version changed from CoCoA-5.3.0 to CoCoA-5.4.0

#22 - 20 Jun 2020 21:25 - John Abbott

- Related to Design #1085: Fns with "OUT" args: should they give ERR::MixedRings? added

#23 - 20 Jun 2020 21:33 - John Abbott

- % Done changed from 30 to 50

I have partly implemented this (in CoCoALib) after discussions wirh Anna (who is uncertain about some aspects).

#24 - 20 Jun 2020 21:34 - John Abbott

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

#25 - 22 Jun 2020 18:00 - John Abbott

- Related to Design #1414: Make class RingElemVector? added

#26 - 28 Sep 2020 16:40 - John Abbott

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

#27 - 22 Oct 2020 17:23 - John Abbott

- Related to Design #1515: Indets in coeffring are ringelems in coeffring? added

#28 - 27 Oct 2020 18:43 - John Abbott

- % Done changed from 50 to 60

I have decided to make this the "master issue" for questions related to automatic ring conversion (mainly because it is the oldest).

Here is a summary of the main points:

- after **use P** all symbols in P are made available as ringelems in P (even if the symbol lies in a subring) -- see #1515 (and #636)
- **no automatic mapping of RingElems for equality tests** (throws MixedRings exception instead) -- see #637
- fns with OUT params (of ringelem type) do not require that the ring of the OUT param be "correct": the right ring will be set by the call (the same way that assignment sets the right ring) -- see #1085 (and #1500 comment 10)

Automatic mapping applies only to single values

- not to matrices (even if 1x1)
- see issue #635
- issue #1493 lets a single value be "promoted" when being assigned to matrix element; but the assignment cannot "promote" the matrix.

#29 - 27 Oct 2020 19:05 - John Abbott

Here are done list and todo list copied from issue #1461 comments 7 & 8. Moving them here.

DONE LIST

- MatrixOps.C
- MatrixOps-KroneckerProd.C
- module.C (mostly)

TODO LIST

- **DenseMat:** myRowMul, myColMul, myAddRowMul, myAddColMul
- FieldIdeal -- needs vector of RingElem in same ring
- **ideal.C** many, some need vector of RingElem in same ring
- **Matrix.C** for SetEntry, also IamEqual
- **MatrixSpecial.C** for JacobianMat which needs vector of RingElem in same ring
- **MatrixView** many cases, some need vector of RingElem in same ring
- PolyFamilies for DicksonPoly
- **ring.C** for operator+= and so on
- **RingElemOps-CoprimeFactorBasis** needs vector RingElem from same ring.
- **SparsePolyOps-XXX** many

#30 - 03 Feb 2022 19:32 - John Abbott

- Target version changed from CoCoA-5.4.0 to CoCoA-5.4.2

#31 - 10 Mar 2023 17:08 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 60 to 80

What is the status of this issue?
I thought it was almost finished.