CoCoA-5 - Design #635

Automatic mapping of RingElem (in operation with a compound value)

22 Oct 2014 20:07 - John Abbott

Status:	Closed	**Start date:**	22 Oct 2014
Priority:	High	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	enhancing/improving	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.4.2	**Spent time:**	2.55 hours

Description

If we decide to allow automatic mapping of RingElem values, Anna suggests that only simple/single values can be automatically mapped.

Thus in a product RingElem*Matrix only the RingElem may be mapped; similarly for operations between RingElem and lists|ideals.

To solve Robbiano's problem we should allow mapping **down** of a RingElem into a subring (if the value lies in the subring).

Discuss!

Related issues:

Related to CoCoA-5 - Feature #7: Automatic mapping between (some) rings	**Resolved**	**20 Oct 2011**
Related to CoCoA-5 - Design #634: Symbol in the coeff ring	**Rejected**	**22 Oct 2014**
Related to CoCoALib - Feature #113: Introduce PartialHom	**In Progress**	**23 Mar 2012**
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 CoCoA-5 - Feature #1461: Automatic mapping for multiplication?	**Closed**	**10 Jun 2020**
Related to CoCoALib - Design #1515: Indets in coeffring are ringelems in coef...	**Rejected**	**22 Oct 2020**
Related to CoCoALib - Feature #645: Automatic mapping of RingElem: user selec...	**Rejected**	**04 Nov 2014**

History

#1 - 22 Oct 2014 20:11 - John Abbott

Robbiano's scenario was the following... a bit simplified:

```
QQab ::= QQ[a,b];
K := NewFractionField(QQab);
Use P ::= K[x,y];
M := mat(K, [[1,2],[3,4]]);
a*M;
```

The result should be a matrix over K; this requires "mapping down" the value of a into the subring K.

#2 - 23 Oct 2014 12:32 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I note that this proposal would still produce an error if one tries to compute:

```
x*M;
```

because x cannot be mapped down into K (and if instead a result were produced, it would have to be a matrix over P).

Is this acceptable? Do we want this?

#3 - 23 Oct 2014 12:57 - Anna Maria Bigatti

The description says it all, but I prefer phrase it also like this:

```
b * M;  -- b RingElem;  M matrix
```

b*M should be **a matrix in the same ring as M**.

Therefore this would not solve situations like

```
x * M;  -- x in QQ[x];  M over QQ
```

because the resulting matrix is not over QQ

This implementation can be done "easily" (so to speak...) but do we want to allow some cases and not others? allow RingElem * matrix, and matrix * RingElem, but not others?

The main problem is that, if we cannot find a general (mathematically robust) definition, we risk to have many exceptions, and to be more confusing than helpful.

#4 - 23 Oct 2014 18:37 - John Abbott

At the moment the idea of automatically "mapping down" ring elements makes me feel uneasy (perhaps solely because I haven't given the matter much consideration up to now).

We must consider all cases of mapping down *e.g.* from R/I to R.

#5 - 11 May 2015 14:35 - John Abbott

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

#6 - 17 Feb 2016 10:42 - Anna Maria Bigatti

- Target version changed from CoCoA-5.1.3/4 Jan 2016 to CoCoA-5.?.?

#7 - 10 Jun 2020 16:07 - Anna Maria Bigatti

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

#8 - 20 Jun 2020 21:32 - John Abbott

- Target version changed from CoCoA-5.?.? to CoCoA-5.4.0

The example in comment 1 works with my current "internal" versions of CoCoA5.

#9 - 22 Oct 2020 17:17 - John Abbott

- % Done changed from 10 to 20

Clarification: the example in comment 1 does indeed run, but the resulting matrix is over P not K.
To obtain a matrix over K, one would have to map the scalar into K (*e.g.* by using LC(a)).

I am tempted to suggest closing this issue since the current impl "works".

#10 - 22 Oct 2020 17:23 - John Abbott

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

#11 - 27 Oct 2020 18:51 - John Abbott

- % Done changed from 20 to 30

I must change the impl so that only the ringelem gets mapped. To map the matrix one must make an explicit call.

One reason for being uneasy about mapping the matrix is that it may be quite costly (if the matrix has many entries).

#12 - 27 Oct 2020 19:11 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 30 to 70

I have changed the impl so that there is no auto promotion for matrices.
The example in comment 1 now gives

```
ERROR: MixedRings: no auto promotion for matrices
```

Perhaps the error could be phrased better?

#13 - 18 Feb 2022 16:43 - Anna Maria Bigatti

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

#14 - 09 Mar 2023 23:11 - John Abbott

- Status changed from Resolved to Feedback

- Assignee set to John Abbott

- % Done changed from 70 to 90

I have changed the err mesg to be more comprehensible: "automatic" instead of "auto".

I think we can close this issue, right?

#15 - 14 Mar 2023 14:46 - Anna Maria Bigatti

John Abbott wrote:

> I have changed the err mesg to be more comprehensible: "automatic" instead of "auto".

what about

```
--> ERROR: MixedRings: no automatic promotion mapping for matrices
```

?

> I think we can close this issue, right?

then yes

#16 - 14 Mar 2023 14:51 - Anna Maria Bigatti

I added the manual search key "promotion mapping" to CanonicalHom and to matrix

#17 - 14 Mar 2023 14:59 - Anna Maria Bigatti

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

Added ".. promotion mapping .." (3 lines) and checked in.
Closing.

#18 - 16 Feb 2024 17:33 - Anna Maria Bigatti

- Related to Feature #645: Automatic mapping of RingElem: user selectable at run-time (GlobalManager?) added