

CoCoA-5 - Design #637

Undesirable consequence of automatic mapping of RingElems?

23 Oct 2014 22:12 - John Abbott

Status:	Closed	Start date:	23 Oct 2014
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	1.81 hour
Target version:	CoCoA-5.4.0	Spent time:	1.80 hour
Description			
Suppose we allow automatic mapping of ringelems into superrings. Consider the following scenario:			
<pre>QQa ::= QQ[a]; use P ::= QQa[x]; A := RingElem(QQa, "a"); // a viewed as elem of QQa B := (x+A)-x; deg(A); --> 1 deg(B); --> 0 A = B; --> true false error?</pre>			
The variables A and B contain "equivalent" values: for instance $(x+A) = (x+B)$ surely gives true. What should the comparison $A = B$ produce?			
Related issues:			
Related to CoCoA-5 - Feature #7: Automatic mapping between (some) rings	Resolved	20 Oct 2011	
Related to CoCoA-5 - Design #636: Distinguish indets from symbols in coeffrin...	Rejected	23 Oct 2014	
Related to CoCoA-5 - Design #635: Automatic mapping of RingElem (in operation...	Closed	22 Oct 2014	
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 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 - 23 Oct 2014 22:23 - John Abbott

- Status changed from New to In Progress
- % Done changed from 0 to 10

I think the values of $\deg(A)$ and $\deg(B)$ are well defined.

If we have $\text{not}(\deg(A) = \deg(B))$ then it would be quite a (nasty?) surprise to have simultaneously $A=B$. Since we have $(x+A) = (x+B)$, it would be a nasty surprise to have simultaneously $\text{not}(A=B)$.

The only option left is for the comparison $A=B$ to produce an error (?mixed rings?).

Automatic mapping means that $A-B$ evaluates to 0. Do we really want two values whose difference is zero to be incomparable???

Help?!?

#2 - 25 Oct 2014 17:06 - John Abbott

Here is a short session valid with the current version of CoCoA-5

```
Fpx := ZZ/(7)[x];
FrF := NewFractionField(Fpx);
A := 2/3;
B := RingElem(FrF,A);
A = B; --> true
num(A);--> 2
num(B);--> 3
```

In other words we have two values which test as equal but which have different numerators. No one has complained loudly about this. Perhaps I should not be so worried about two equal values having different degrees?

What do you think?

#3 - 07 Nov 2014 16:49 - John Abbott

While talking to Anna the following idea came up. Non-simple elements of coefficient rings are printed inside brackets, except if the element is the coefficient of 1 in a polynomial ring. We could change the rules for using brackets so that elements of subrings are always printed in brackets (incl. elements of finite fields?). Thus in the example in the preceding comment `num(B)` would print out as (3) rather than a plain 3; the idea is that the brackets warn the user that the printed value is not exactly as it seems.

Similarly in the example given in the description the two printed values would be different: one without brackets, namely just `a`, and the other with, namely `(a)`.

These brackets could be useful for informing the reader that the value is not "plain", but it might also "uglify" printing in general (when perhaps distinguishing the cases is not so important). In any case the `RingOf` function can always be used to find out where the value resides.

Comments? Criticisms?

#4 - 11 May 2015 14:34 - John Abbott

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

#5 - 17 Feb 2016 10:43 - Anna Maria Bigatti

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

#6 - 10 Jun 2020 16:07 - Anna Maria Bigatti

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

#7 - 08 Oct 2020 13:48 - John Abbott

- *Target version changed from CoCoA-5.?.? to CoCoA-5.4.0*
- *% Done changed from 10 to 20*

I have just tried the example in the description. The equality test gives ERR::MixedRings, which I think is probably "correct".

#8 - 22 Oct 2020 17:08 - John Abbott

- *% Done changed from 20 to 30*

I now suggest that we make equality-test throw an error (MixedRings) if the args are elements of two different rings.

If it later turns out to be awkward, we can revise the design -- not strictly backward compatible (but "removing an error case" could be viewed as progress).

Strictly this issue should be under CoCoALib...

#9 - 22 Oct 2020 17:29 - John Abbott

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

#10 - 27 Oct 2020 18:15 - John Abbott

- *Status changed from In Progress to Closed*
- *Assignee set to John Abbott*
- *% Done changed from 30 to 100*
- *Estimated time set to 1.81 h*

Conclusion: no automatic mapping of ringelems in equality tests

A user working in several rings at once has to some responsibility for handling things correctly; it seems too hard to "read the user's mind". The safe option seems to be to throw an exception... hopefully experience will confirm this decision.

PS Anna agreed while discussing this.

#11 - 16 Feb 2024 17:32 - Anna Maria Bigatti

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