

CoCoALib - Design #1515

Indets in coeffring are ringelems in coeffring?

22 Oct 2020 17:23 - John Abbott

Status:	Rejected	Start date:	22 Oct 2020
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Tidying	Estimated time:	0.99 hour
Target version:	CoCoALib-0.99800	Spent time:	0.95 hour
Description			
<p>A suggestion (but maybe not a good one).</p> <p>Now that we have some degree of automatic ringelem conversion, we could say that indets in a coeffring are actually ringelems in the coeffring.</p> <p>Currently we have:</p> <pre>P1 ::= QQ[a]; P2 ::= P1[x]; use P2; RingOf(a) = P2; --> true</pre> <p>Might it make sense to have the following?</p> <pre>P1 ::= QQ[a]; P2 ::= P1[x]; use P2; RingOf(a) = P1; --> true</pre>			
Related issues:			
Related to CoCoA-5 - Feature #7: Automatic mapping between (some) rings		Resolved	20 Oct 2011
Related to CoCoA-5 - Design #635: Automatic mapping of RingElem (in operation...		Closed	22 Oct 2014
Related to CoCoA-5 - Design #637: Undesirable consequence of automatic mappin...		Closed	23 Oct 2014
Related to CoCoA-5 - Design #636: Distinguish indets from symbols in coeffrin...		Rejected	23 Oct 2014

History

- #1 - 22 Oct 2020 17:23 - John Abbott
- Related to Feature #7: Automatic mapping between (some) rings added
- #2 - 22 Oct 2020 17:23 - John Abbott
- Related to Design #635: Automatic mapping of RingElem (in operation with a compound value) added
- #3 - 22 Oct 2020 17:29 - John Abbott
- Category set to Tidying
 - Target version set to CoCoALib-0.99850

The proposed change might be inconvenient in view of issue [#637](#): the equality test $x = a$ would throw a MixedRings exception.

It is also potentially unhelpful if we use a fraction field:

```
P1 ::= QQ[a];
FrF := NewFractionField(P1);
```

```
P2 ::= FrF[x];  
use P2;  
RingOf(a); --> P1, FrF or P2?  
x+1/a; --> this should work, and *not* complain that 1/a cannot be computed
```

Mmmm, now the proposal is looking like a **bad idea** :-)

#4 - 22 Oct 2020 17:29 - John Abbott

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

#5 - 22 Oct 2020 17:31 - John Abbott

- Related to Design #636: Distinguish indets from symbols in coeffring in Use command added

#6 - 22 Oct 2020 17:32 - John Abbott

This is almost a duplicate of issue [#636](#).

#7 - 23 Oct 2020 09:01 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 20

I am increasingly convinced that this is not a good idea. The example in comment 3 is already fairly convincing. In that example there are 3 rings all containing a: namely, P1, FrF and P2. Which to use when the user types a?

I now think that the main point is that user said explicitly **use P2** -- this sets the **CurrentRing** (whose main purpose is to specify how user inputs are interpreted).

If the user really wants a in some other ring then it is not so hard to write RingElem(FrF, "a")... though, admittedly, this is cumbersome (and not obvious to a novice).

In summary: I think CoCoA should respect the **use** command. The advantage is that it is clear how CoCoA will interpret input. Also functions such as RingElem and LC (or ConstantCoeff) let the user easily specify other "closely related" rings... but I do accept that occasionally it might be nicer if CoCoA could "read the user's mind".

I propose rejecting this idea (incl. in other issues where it has been proposed).

#8 - 27 Oct 2020 18:01 - John Abbott

- Status changed from In Progress to Rejected

- Assignee set to John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99800

- % Done changed from 20 to 100

- Estimated time set to 0.99 h

I am rejecting this because of the example in comment 3 above.

I have also rejected the duplicate issue [#636](#).