

CoCoA-5 - Design #634

Symbol in the coeff ring

22 Oct 2014 16:08 - John Abbott

Status:	Rejected	Start date:	22 Oct 2014
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	0.00 hour
Target version:	CoCoA-5.1.2 summer 2015	Spent time:	1.00 hour
Description			
Consider the following mini-session			
<pre>QQab ::= QQ[a,b]; K := FractionField(QQab); Use P ::= K[x,y]; a; // in which ring???</pre>			
In which ring should the value of a lie? Currently, it lies in the ring P so that expressions such as x+a are accepted (<i>i.e.</i> no problem of <i>mixed rings</i>).			
If we allow automatic mapping from a subring to a superring then we could put the value of a in the smallest subring which contains it (in this case K), then in the expression x+a the value of a is automatically mapped into P before being added to x.			
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 #636: Distinguish indets from symbols in coeffrin...		Rejected	23 Oct 2014

History

#1 - 22 Oct 2014 16:11 - John Abbott

The problem was highlighted when Robbiano wanted to multiply a matrix (over K) by a; CoCoA-5 gave a *mixed rings* error. A simpler case would be a*LC(x+a); this will currently produce the same error, yet many would expect it to work.

#2 - 22 Oct 2014 18:35 - John Abbott

There is a potential problem with my suggestion of symbols being mapped into the smallest containing subring. In the example I gave above, one would expect the expression 1/a to be evaluated without problem, but this won't happen if a is mapped into QQab.

So it would seem that I'd want the expression to be computed in the "smallest subring where the expression is valid" (or error if there is no such subring); this in turn would imply that the current ring would have to be passed as a parameter (so we know which subrings to consider)... I do not much like this. Consider continuing the session as follows:

```
x+1/a; // this *MUST* work
1/a; // this really should work too
a2 := a;
Use QQ[r,s];
1/a2; // what happens here?
```

If a2 lies in the smallest subring then its reciprocal cannot be computed, and it is not so obvious how to find the right superring to map it into.

Trickier than I thought...

#3 - 22 Oct 2014 20:02 - John Abbott

- Status changed from New to Rejected
- Assignee set to John Abbott
- % Done changed from 0 to 100

My previous comment convinces me that it is not feasible to make a have the value of the symbol a in the smallest subring. Furthermore it seems risky to me to change the current approach. For instance, one might want to perform the following:

```
deg(a);
```

How should CoCoA behave?

1. it produces 0 (considering a as an element of P)
2. it produces 1 (considering a as an element of $QQab$)
3. it produces an error (considering a as an element of K)

I believe that **3** is the least helpful. I can see arguments in favour of both **1** and **2**, though my preference definitely lies with **1** because of the following scenario:

```
a3 := (x+a)-x; // really just a
deg(a3); // surely gives 0
a3 = a; // gives true, I hope!
```

It would be quite a **nasty surprise** if the equality test gave any result other than true; and it would then be a nasty surprise if $\text{deg}(a) = \text{deg}(a3)$ gave any result other than true.

I am now convinced that this idea should be rejected.