

CoCoA-5 - Design #546

ideal wants LIST of RINGELEM

05 May 2014 15:07 - John Abbott

|  |                     |                        |             |
|--|---------------------|------------------------|-------------|
| <b>Status:</b>   | Closed              | <b>Start date:</b>     | 05 May 2014 |
| <b>Priority:</b>   | Normal              | <b>Due date:</b>       |             |
| <b>Assignee:</b>   | Anna Maria Bigatti  | <b>% Done:</b>         | 100%        |
| <b>Category:</b>   | Cleaning            | <b>Estimated time:</b> | 0.90 hour   |
| <b>Target version:</b>   | CoCoA-5.1.1 Seoul14 | <b>Spent time:</b>     | 0.80 hour   |
| <b>Description</b><br>It is inconvenient (perhaps even embarassing) that you cannot do this:<br><br>Use QQ[x,y,z];<br>I := ideal([x,0,y]);<br><br>It really should be able to convert the zero (or any rational number) into the appropriate ring, so long as there is at least 1 RINGELEM value.<br><br>[I bet this is related to some other issue... no time to check now] |                     |                        |             |
| <b>Related issues:</b><br>Related to CoCoA-5 - Feature #453: Automatic conversion from INT (or RAT) to ...<br><div>In Progress03 Mar 2014</div>  |                     |                        |             |

History

#1 - 22 Jul 2014 08:27 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti
- % Done changed from 0 to 50

It was easy to fix it for ideal([x,0,y]);.  
I re-designed the function for ideal a bit.  
I renamed evalArgAsRingElemList into evalArgAsListOfRingElem (easier to find and to relate it to the less flexible evalArgAsListOf<RingElem>)  
  
Much more tedious to do for ideal(x,0,y);. Postpone? Ignore?

#2 - 02 Sep 2014 11:08 - John Abbott

- Status changed from New to Closed
- % Done changed from 50 to 100
- Estimated time set to 0.90 h

After discussing with Anna we have decided to accept the current solution: i.e. that ideal([x,0,y]) works as desired but ideal(x,0,y) does not. Fixing the latter looks to be quite tricky, and in any case the problem has an easy workaround: just put the generators into a list!