

CoCoA-5 - Feature #1003

New syntax for creating poly rings?

27 Jan 2017 00:40 - John Abbott

Status:	In Progress	Start date:	27 Jan 2017
Priority:	Normal	Due date:	
Assignee:		% Done:	10%
Category:	enhancing/improving	Estimated time:	0.00 hour
Target version:	CoCoA-5.?.?	Spent time:	0.55 hour
Description			
Here is an idea which could make creating poly rings a "normal case" rather than a "special case".			
Instead of having to use the special operator <code>*::=</code> to allow a special syntax:			
<pre>P ::= QQ[x,y,z];</pre>			
we could use normal syntax if the indet names were inside a string:			
<pre>P := QQ["x,y,z"];</pre>			
Related issues:			
Related to CoCoA-5 - Design #997: Using protected variable names for "bound v...		Closed	18 Jan 2017
Related to CoCoALib - Feature #1330: New syntax for NewQuotientRing		Closed	08 Oct 2019
Related to CoCoA-5 - Support #1418: Manual entry for NewPolyRing		New	15 Feb 2020
Related to CoCoA-5 - Feature #657: use command, ring syntax, RingOf		New	20 Jan 2015
Related to CoCoA-5 - Feature #1503: More flexible ring creation syntax (after...		New	08 Oct 2020

History

#1 - 27 Jan 2017 00:46 - John Abbott

Some advantages are:

- do not need operator `::=` to introduce special syntax (valid only in a special context)
- this would allow expressions such as `QQ["x,y,z"]` to be placed in normal formulas, for instance as args to a fn call `ComputeResultIn(QQ["a,b,c"])`

Some disadvantages are:

- not as natural as the current special syntax (because you need to use quotes)
- not clear how the term ordering would be specified

There are some further matters to be decided: if I write `QQ["x,y,z"]` twice, will that produce the same poly ring (JAA: probably it should -- achieving this may be not entirely straightforward).

#2 - 27 Jan 2017 00:49 - John Abbott

One problem it does not solve is how to write something like `QQ[alpha]/(alpha^2-2)` since we cannot create the ideal generated by `alpha^2-2` until the ring has been successfully built.

#3 - 27 Jan 2017 07:40 - Anna Maria Bigatti

- % Done changed from 0 to 10

Neat idea, but I think we would still have the ambiguity between:
 $K[x]$ and $F[\text{"factors"}]$ for records (very useful for making loops on the fields).

Anyway this is not a problem as there is for $K[x]$, where x is an undefined token for the interpreter.

Another ambiguity is
 $X := "a,b,c"; \text{ use } QQ[X];$

Even though I agree that $::=$ is confusing, I think this new syntax may cause more confusion. And I'd rather concentrate on finding a pretty syntax for quotient rings.

#4 - 27 Jan 2017 07:42 - Anna Maria Bigatti

Remember the syntax `NewPolyRing(QQ, "x,y,z");`. That's very expressive!

#5 - 27 Jan 2017 15:54 - John Abbott

- Status changed from New to In Progress

I do not believe that there will be ambiguity: consider the expression **OBJ**[string]

- if **OBJ** is a record then it is clear what to do
- if **OBJ** is a ring then we build a polynomial ring
- otherwise error

Note that **OBJ**[int] already has three meanings:

- if **OBJ** is a list then get the corresponding entry
- if **OBJ** is an INTMAP then get the corresponding entry
- if **OBJ** is a matrix then get the corresponding row
- otherwise error

#6 - 27 Jan 2017 15:59 - John Abbott

- Related to Design #997: Using protected variable names for "bound variables" (e.g. for, try...endtry) added

#7 - 15 May 2020 10:39 - Anna Maria Bigatti

- Related to Feature #1330: New syntax for NewQuotientRing added

#8 - 15 May 2020 10:41 - Anna Maria Bigatti

- Related to Support #1418: Manual entry for NewPolyRing added

#9 - 08 Oct 2020 13:55 - John Abbott

- Related to Feature #657: use command, ring syntax, RingOf added

#10 - 08 Oct 2020 14:02 - John Abbott

- Related to Feature #1503: More flexible ring creation syntax (after use or ::=) added