

## CoCoA-5 - Design #1327

### Inconvenient PolyRingHom, PolyAlgebraHom

07 Oct 2019 17:13 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	07 Oct 2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	enhancing/improving	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CoCoA-5.3.0	<b>Spent time:</b>	1.25 hour
<b>Description</b>			
<p>I wanted to do in CoCoA-5 an example/exercise from Martin Kreuzer's lecture course on computer algebra. It turned out to be more awkward than I expected.</p> <p>Can we make it easier?</p> <p>The example was simply to compute the kernel of the algebra hom from <math>\mathbb{Q}\mathbb{Q}[z]</math> which maps <math>z</math> to <math>(1+\sqrt{5})/2</math>.</p> <p>The problem I had was finding a neat way to create the hom (incl. having to look things up in the manual).</p> <p><b>2020-02</b> solution:</p> <pre>/**/ phi := PolyAlgebraHom(NewPolyRing(QQ, "x"),                           NewQuotientRing(NewPolyRing(QQ, "y"), "y^2-5"),                           "(1+y)/2"); /**/ ker(phi);</pre>			
<b>Related issues:</b>			
Related to CoCoALib - Feature #1330: New syntax for NewQuotientRing		<b>Closed</b>	<b>08 Oct 2019</b>
Related to CoCoALib - Feature #1329: New syntax for creating homomorphisms (P...		<b>Closed</b>	<b>08 Oct 2019</b>

#### History

##### #1 - 07 Oct 2019 17:25 - John Abbott

Here is the solution I ended up producing, but I found discovering it to be unnatural.

```
>>> use QQy := QQ[y];
>>> f := y^2-5;
>>> QQx := QQ[x];
>>> R := NewQuotientRing(QQy, ideal(f));
>>> use R;
>>> ?induced
>>> phi := PolyAlgebraHom(QQx, QQy, [indet(QQy, 1)]);
>>> PolyRingHom(QQx, RHS, IdentityHom(QQ), [(1+y)/2]);
--> ERROR: RingHom has wrong codomain
>>> ?PolyRingHom
>>> phi := PolyRingHom(QQx, RHS, CanonicalHom(QQ, RHS), [(1+y)/2]);
>>> ker(phi);
ideal(x^2 -x -1)
```

Actually, I now see that my first attempt to use PolyAlgebraHom failed because I had the wrong CurrentRing. So this shorter approach also works... perhaps I was just being stupid/error-prone...

```
>>> use QQy := QQ[y];
>>> f := y^2-5;
>>> R := NewQuotientRing(QQy, ideal(f));
>>> use R;
>>> QQx := QQ[x];
>>> phi := PolyAlgebraHom(QQx, R, [(1+y)/2]);
>>> ker(phi);
```

```
ideal(x^2 -x -1)
```

## #2 - 08 Oct 2019 08:57 - Anna Maria Bigatti

- % Done changed from 0 to 50

Alternative, without use:

```
QQy := QQ[y];  
R := NewQuotientRing(QQy, ideal(RingElem(QQy, "y^2-5")));  
QQx := QQ[x];  
phi := PolyAlgebraHom(QQx, R, [RingElem(R, "(1+y)/2")]);  
ker(phi);
```

add this to the manual?

close this issue?

## #3 - 08 Oct 2019 10:35 - John Abbott

- Status changed from New to In Progress

If we find it non-trivial to create a good solution then we really should include some example(s) in the documentation!

It would be nice if we could produce an easier way to make quotient rings. For example something like this:

```
R ::= QQ[y]/(y^2-5);  
QQx := QQ[x];  
phi := PolyAlgebraHom(QQx, R, [RingElem(R, "(1+y)/2")]);  
ker(phi);
```

It would be still nicer if we could specify rings inside other expressions: something a bit like this...

```
phi := PolyAlgebraHom(QQ[x], QQ[y]/(y^2-5), ["(1+y)/2"]);  
ker(phi);
```

Of course, as it is written here, it is "ambiguous". :-/

#### #4 - 08 Oct 2019 12:03 - Anna Maria Bigatti

- Priority changed from Low to Normal

John Abbott wrote:

It would be nice if we could produce an easier way to make quotient rings. For example something like this:  
[...]

It would be still nicer if we could specify rings inside other expressions: something a bit like this...  
[...]  
Of course, as it is written here, it is "ambiguous". :-/

Nice idea!! Making a new issue.

#### #5 - 08 Oct 2019 12:20 - Anna Maria Bigatti

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

#### #6 - 08 Oct 2019 12:21 - Anna Maria Bigatti

- Related to Feature #1329: New syntax for creating homomorphisms (PolyAlgebraHom) added

#### #7 - 10 Oct 2019 09:42 - Anna Maria Bigatti

- Status changed from In Progress to Resolved

- Assignee set to Anna Maria Bigatti

- % Done changed from 50 to 80

The power of the new string syntax: can be defined in one line, and the strings are evaluated in the appropriate ring, with no ambiguity. Moreover we can write "y<sup>2</sup>-5" when QQ[y] does not yet exist.

```
phi := PolyAlgebraHom(NewPolyRing(QQ, "x"),  
                    NewQuotientRing(NewPolyRing(QQ, "y"), "y^2-5"),  
                    "(1+y)/2");
```

Works also in CoCoALib.

#### #8 - 10 Oct 2019 09:46 - Anna Maria Bigatti

- Target version changed from CoCoA-5.?.? to CoCoA-5.3.0

#### #9 - 10 Oct 2019 18:41 - Anna Maria Bigatti

- Status changed from Resolved to Feedback
- % Done changed from 80 to 90

**#10 - 13 Feb 2020 09:35 - Anna Maria Bigatti**

- Description updated
- Status changed from Feedback to Closed
- % Done changed from 90 to 100