CoCoALib - Feature #1532

Convenient RingHom pseudo-ctors

06 Nov 2020 13:14 - John Abbott

Status:	In Progress	**Start date:**	06 Nov 2020
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	10%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99900	**Spent time:**	0.95 hour

Description

CoCoALib should offer some convenient pseudo-ctors for RingHom.

One example would be from QQ[x,y,z] into QQ(x)[y,z]; where indets names are to be preserved.

Prompted by a question from Florian Walsh.

Related issues:

Related to CoCoA-5 - Design #1533: RingElem and IndetSymbols not compatible	**In Progress**	**06 Nov 2020**
Related to CoCoALib - Feature #1329: New syntax for creating homomorphisms (P...	**Closed**	**08 Oct 2019**

History

#1 - 06 Nov 2020 13:20 - John Abbott

Here is the example Florian wanted:

```
P1 ::= QQ[x,y,z];
P2 ::= QQ(x)[y,z]; --> cannot do this directly
phi := PolyAlgebraHom(P1,P2, [RingElem(P2, name) | name in ["x","y","z"]]);
```

It would be nice to have an easy way to make such ringhoms.

Probably we should try examples before aiming to write down exact semantics.

#2 - 06 Nov 2020 13:35 - John Abbott

- Related to Design #1533: RingElem and IndetSymbols not compatible added

#3 - 06 Nov 2020 14:07 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Here is a prototype in CoCoA-5... we would also need a version for cocoalib!!

```
define AllSymbols(R)
  if IsZZ(R) or IsQQ(R) then return []; endif;
  if IsPolyRing(R) then return concat(AllSymbols(CoeffRing(R)), IndetSymbols(R)); endif;
```

```
  if IsQuotientRing(R) then return AllSymbols(BaseRing(R)); endif;
  if IsFractionField(R) then return AllSymbols(BaseRing(R)); endif;
  println "How did we get here? R = ", R;
  error("surprise!");
enddefine; -- AllSymbols

define NaturalHom(P1,P2)
  if not(IsPolyRing(P1)) then error("NaturalHom: not a polyring -- giving up!"); endif;
  Symbols1 := AllSymbols(P1);
  Symbols2 := AllSymbols(P2);
  CommonSymbols := intersection(Symbols1, Symbols2);
  if CommonSymbols = [] then error("NaturalHom: no common symbols"); endif;
  ExtraSymbols1 := diff(Symbols1, CommonSymbols);
  images := [];
  foreach symb in Symbols1 do
    if symb isin ExtraSymbols1 then
      append(ref images, zero(P2));
    else
      append(ref images, RingElem(P2, concat([symb.head], symb.indices)));
    endif;
  endforeach;
  return PolyAlgebraHom(P1, P2, images);
enddefine; -- NaturalHom
```

#4 - 06 Nov 2020 14:24 - John Abbott

Florian mentions that Singular has something called **imap**...

#5 - 09 Nov 2020 10:03 - Anna Maria Bigatti

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

#6 - 10 Nov 2020 14:11 - John Abbott

It might be possible to extend the impl give above to handle quotients of poly rings, and perhaps also "towers" of poly rings (or are they already handled?)

What about the case QQ[x,y] to ZZ/(p)[x,y]? It would be convenient (but is also unclean).

#7 - 07 Mar 2024 21:34 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99900