CoCoALib - Design #1223

NewPolyRing default indet names

02 Sep 2018 21:32 - John Abbott

Status:	Closed	**Start date:**	02 Sep 2018
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Safety	**Estimated time:**	4.01 hours
Target version:	CoCoALib-0.99650 November 2019	**Spent time:**	4.00 hours

Description

Currently NewPolyRing may be called specifying just the number of indets (but not their names).
Currently the names chosen are x[1] up to x[n].

This has caused trouble in the factorizer of algebraic extensions.

JAA thinks it would be better if the ctor chose "anonymous" indet names (this would surely avoid all possible name clashes, incl. the one discovered in the factorizer over algebraic extensions).

An alternative could be to eliminate the ctors which do not require a list of indet names: the caller could then explicitly create a list if indet names using NewSymbols or SymbolRange.

Discuss; decide; implement.

Related issues:

Related to CoCoALib - Feature #320: PPMonoid pseudo-ctors without symbol names	**Rejected**	**21 Feb 2013**
Related to CoCoA-5 - Support #1240: John's visit Feb 2019	**Closed**	**08 Feb 2019**

History

#1 - 02 Sep 2018 21:33 - John Abbott

- Related to Feature #320: PPMonoid pseudo-ctors without symbol names added

#2 - 02 Sep 2018 21:35 - John Abbott

The problem was originally communicated to JAA via email. The following is a failing example:

```
auto Q = CoCoA::RingQQ();
auto Qx = CoCoA::NewPolyRing(Q, CoCoA::symbols("x"));
auto x = CoCoA::indets(Qx)[0];
auto cf = x*x-2;
auto Qadjx = CoCoA::NewQuotientRing(Qx, CoCoA::ideal(cf));
auto Qy = CoCoA::NewPolyRing(Qadjx, CoCoA::symbols("y"));
auto y = CoCoA::indets(Qy)[0];
auto cg = y*y - 3;
auto Qadjy = CoCoA::NewQuotientRing(Qy, CoCoA::ideal(cg));
auto Qz = CoCoA::NewPolyRing(Qadjy, CoCoA::symbols("z"));
auto z = CoCoA::indets(Qz)[0];
auto ch = z*z*z*z-4;
std::cout << "factor(" << ch << ") = " << CoCoA::factor(ch) << std::endl;
```

The call to factor (in the last line) fails and throws an error about unsuitable indet name.

#3 - 02 Sep 2018 21:41 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The problem arises in a call to **Hom_P_Px** which creates a new polyring by calling **NewPolyRing** specifying only the number of indets (but not their names). This pseudo-ctor call wants to use names x[1], x[2] and so on, but this clashes with the x used to build the first alg.extn.

The problem can be fixed by changing to Hom_P_Px so that it creates an explicit list of indet names using NewSymbols.

BUT JAA thinks that the problem lies in the pseudo-ctor NewPolyRing which accepts only the number of indets, and which currently uses the indet names x[1], x[2] and so on. JAA thinks that the names should either be produced by NewSymbols, or that the pseudo-ctor should be eliminated.

#4 - 04 Sep 2018 14:33 - Anna Maria Bigatti

We first had this "friendly" pseudo constructor because the syntax for indet names was tedious.
Now we have SymbolRange, and NewPolyRing(RingQQ(), symbols("x,y,z")) which is even nicer.
(maybe we should even allow NewPolyRing(RingQQ(), "x,y,z")?)

So, after defending them for ages, I'm now in favour of removing the pseudo constructors taking just the number of indets.

#5 - 28 Sep 2018 14:51 - John Abbott

I see two possible actions:
(A) remove completely the pseudo-ctor which accepts just the number of indets; the caller must then supply indet names in the call *e.g.* NewPolyRing(QQ, NewSymbols(nvars))
(B) keep the ctor, and make it use "anonymous" symbols instead of x[1], x[2], etc

An advantage of **(A)** is that it is clear at the point of the call that anon symbols will be used.
A disadvantage of **(A)** is that the call itself is longer (since it must call NewSymbols(nvars).
The advantage of **(B)** is that it is backward compatible (but how many times is NewPolyRing(QQ,nvars) called?)

Do you prefer **(A)** or **(B)** ? Opinions?

#6 - 28 Sep 2018 14:57 - Anna Maria Bigatti

John Abbott wrote:

> Do you prefer **(A)** or **(B)** ? Opinions?

I prefer A, as I had already explained.

#7 - 28 Sep 2018 18:05 - John Abbott

- Assignee set to John Abbott

- % Done changed from 10 to 60

I have updated the code: sources, tests and examples.
Everything compiles, and the tests pass (after changing some expected outputs).
I have checked in -- have fun, Anna!

Still have to change the doc... And removed the commented out old impls.

#8 - 28 Sep 2018 20:36 - Anna Maria Bigatti

Just to remember.
This should go into obsolescent.

#9 - 29 Sep 2018 12:51 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 60 to 70

Removed the doc. Moved the old fns into obsolescent.

#10 - 26 Feb 2019 16:26 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 70 to 90

This has been "resolved" for 5 months. Moving to "feedback". Hope to close while I'm in Genoa.

#11 - 26 Feb 2019 16:26 - John Abbott

- Related to Support #1240: John's visit Feb 2019 added

#12 - 04 Mar 2019 18:04 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

#13 - 10 Oct 2019 19:13 - Anna Maria Bigatti

- Estimated time set to 4.01 h