CoCoALib - Design #829

PPOrderingCtor: name of mem fn which actually constructs?

30 Nov 2015 17:48 - John Abbott

Status:	Closed		**Start date:**	30 Nov 2015
Priority:	High		**Due date:**	
Assignee:	John Abbott		**% Done:**	100%
Category:	Tidying		**Estimated time:**	2.34 hours
Target version:	CoCoALib-0.99560		**Spent time:**	2.45 hours

Description

Currently PPOrderingCtor has a mem fn called myCtor which accepts an integer arg and returns the ordering for than number of indets.

Might it be better to use operator() instead of myCtor?

Currently we do this

```
PPOrderingCtor ord = lex;
...
PPOrdering O = ord.myCtor(NumIndets);
```

Using operator() would let us write this more compact version:

```
PPOrderingCtor ord = lex;
...
PPOrdering O = ord(NumIndets);
```

It is more compact, but is it clearer? Or less clear? Does it matter?

History

#1 - 30 Nov 2015 17:52 - John Abbott

We could also have both, but I do not usually like to offer two ways of achieving the same thing... a user might be puzzled as to why both exist (unless the documentation simply says they are equivalent and the user can choose whichever syntax pleases more).

In fact, I doubt any normal user would want to operate directly on objects of type PPOrderingCtor; it was just a "trick" to allow the user to write lex, StdDegRevLex etc as an ordering when creating a poly ring (or a PPMonoid).

#2 - 30 Nov 2015 18:06 - Anna Maria Bigatti

Would it be possible to write PPOrdering O = lex(NumIndets);?
If not, I fear it might be more confusing.
In fact I would not call the variable ord,but ordctor,then the function should be ordctor. myMakeOrd(n) ....
... Ugly, I know...

#3 - 30 Nov 2015 18:23 - John Abbott

Yes, it should be possible to write lex(NumIndets), but normally it would be enough to write just lex and let the called fn actually invoke the

pseudo-ctor with the correct number of indets.

PPMonoidEv(symbols("x,y,z"), lex)
is equivalent to
PPMonoidEv(symbols("x,y,z"), lex(3))
just in the first version you do not need to worry about getting the number of indets right (since the call to PPMonoidEv does that automatically).

#4 - 30 Nov 2015 18:27 - Anna Maria Bigatti

Ok, good for operator()!
Still I 'd call the variable OrdCtor ;-)

#5 - 01 Dec 2015 13:55 - John Abbott

- Status changed from New to Resolved

- Assignee set to John Abbott

- % Done changed from 0 to 80

I've made the change (plus some very minor cleaning), incl changing the arg names!

Also made the analogous change for ModuleOrderingCtor.

Will change doc, then check in.

#6 - 01 Dec 2015 14:04 - John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99540 Feb 2016

There are two ways to create a "lex" ordering for N indets:
NewLexOrdering(N) or lex(N).

I wonder whether we should simply eliminate the fn NewLexOrdering? What do you think?

Of course, the same applies to StdDegLex and StdDegRevLex.

NewMatrixOrdering is however different!

#7 - 23 Mar 2016 15:29 - Anna Maria Bigatti

- Priority changed from Low to High

- Target version changed from CoCoALib-0.99540 Feb 2016 to CoCoALib-0.99550 spring 2017

#8 - 22 Apr 2017 23:19 - John Abbott

Anna, you have not answered my question in comment 6: whether to make obsolescent NewLexOrdering

#9 - 28 Apr 2017 09:32 - Anna Maria Bigatti

- Estimated time set to 1.51 h

#10 - 28 Apr 2017 15:17 - Anna Maria Bigatti

John Abbott wrote:

> NewLexOrdering(N) or lex(N).
> I wonder whether we should simply eliminate the fn NewLexOrdering? What do you think?

yes!

#11 - 08 May 2017 11:53 - Anna Maria Bigatti

- Target version changed from CoCoALib-0.99550 spring 2017 to CoCoALib-0.99560

I see now that this was not done for the release.
Postpone to next.

#12 - 10 Nov 2017 15:09 - John Abbott

- Status changed from Resolved to Closed

- % Done changed from 80 to 100

- Estimated time changed from 1.51 h to 2.34 h

Eliminated NewLexOrdering, NewStdDegLexOrdering, NewStdDegRevLexOdering.
Made all consequent changes. Changed doc.

Compiles, and all tests pass.