CoCoALib - Slug #679

power for PPs is too slow

13 Apr 2015 14:54 - John Abbott

Status:	Closed	**Start date:**	13 Apr 2015
Priority:	High	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Safety	**Estimated time:**	2.56 hours
Target version:	CoCoALib-0.99536 June 2015	**Spent time:**	2.55 hours

Description

Profiling showed that powering PPs incurred a significant time penalty in the (prototype) code for ImplicitDirect.

The problem is that the default repr for exponents is "order vector", and checking for overflow is costly in this case (as myDecompress must be called).

Proposal: separate the overflow checking from the computation of the power.

Related issues:

Related to CoCoALib - Feature #269: PPMonoids: check for exponent overflow in...	**Closed**	**18 Oct 2012**
Related to CoCoA-5 - Bug #267: Cyclotomic(106743) fails	**Closed**	**18 Oct 2012**
Related to CoCoALib - Slug #742: View PP exponent vector (and order vector?) ...	**In Progress**	**28 Jun 2015**

History

#1 - 13 Apr 2015 14:59 - John Abbott

- Assignee set to John Abbott

The proposed design is for each PPMonoid to offer 2 fns:

- myPower which just computes the power **without checking** for overflow
- myPowerOverflowCheck which just checks for overflow **without computing** the power

With Anna we had considered the following design:

- myPowerFast compute power, no overflow check
- myPowerCheck compute power, with overflow check

However, in the end I preferred the former design: it fits better with the design motto of making a fn do just 1 task, and seems easier to use in practice.

#2 - 13 Apr 2015 15:01 - John Abbott

- % Done changed from 0 to 30

I have implemented a first version. All tests pass. Still needs cleaning; must update doc. I'll pass prototype to Anna so she can verify it solves the speed problem.

Update Anna reports a slight improvement in speed (but much less than hoped).

#3 - 28 Jun 2015 19:34 - John Abbott

- % Done changed from 30 to 60

Apparently I have already done almost everything (but still have to check in).

I was surprised to see that we do not do anything "clever" when multiplying or powering: I had expected the code to view the exponent vectors as an array of long with the possibility of handling several exponents at the same time. Is this a bug or a feature?

#4 - 29 Jun 2015 13:02 - John Abbott

- Status changed from New to Closed

- % Done changed from 60 to 100

Apparently everything was already checked in... So I'm considering that ad hoc in-house testing has already happened... Closing this issue!

#5 - 29 Jun 2015 13:03 - John Abbott

- Estimated time set to 2.56 h