CoCoALib - Design #778

CRTMill::myAddInfo accept modulus 1 or not?

17 Sep 2015 20:16 - John Abbott

Status:	In Progress	**Start date:**	17 Sep 2015
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Tidying	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	0.60 hour

Description

Should the mem fn CRTMill::myAddInfo accept a residue-modulus pair with modulus equal to 1?
Such a pair actually adds no information at all. Currently, normal mode accepts it, but debugging mode produces an "assertion failure".

Which behaviour is better?

Related issues:

Related to CoCoALib - Design #620: Redesign CRTMill	**In Progress**	**11 Sep 2014**

History

#1 - 17 Sep 2015 20:21 - John Abbott

JAA found the bug when trying to run a CRT loop in CoCoA-5. The initial CRT residue-modulus pair was naturally (0,1); then the very first call to CRT (in CoCoA-5) produced an assertion failure because CRTMill::myAddInfo as called with one input modulus being 1 (and by chance JAA had compiled with debugging active).

It would certainly be convenient to accept modulus 1 (and the residue is completely ignored) so that CRT loops can easily be written.

Adding a residue-modulus pair with modulus=1 adds no information (but is also quite harmless); it would probably be a mistake except as the very first pair.

#2 - 17 Sep 2015 20:26 - John Abbott

- Status changed from New to In Progress

The possible approaches are:
(A) always give error when an input modulus is 1 (but awkward for starting a CRT loop)
(B) always accept an input modulus of 1 (pointless but harmless; current behaviour)
(C) accept modulus of 1 only if internal modulus is 1 (i.e. just as the first pair)

Approach (C) is tempting, but it does mean that in CoCoA-5 a call like CRT(0,1,A,B) works as expected while CRT(A,B,0,1) would produce an error (with the current impl of CRT...)

Comments? Ideas?

#3 - 21 Mar 2016 15:15 - John Abbott

- Related to Design #620: Redesign CRTMill added

#4 - 21 Mar 2016 15:16 - John Abbott

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