CoCoALib - Feature #627

Gaussian integer and rationals ZZi, QQi

22 Sep 2014 18:15 - John Abbott

Status:	New	**Start date:**	22 Sep 2014
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-1.0	**Spent time:**	0.50 hour

Description

Ulrich would like to have an easy way to compute with (approximations to) complex numbers.

Currently, it is possible to create QQ[i] and ZZ[i] as quotients of polynomial extensions, but the procedure is cumbersome, and the resulting ring could be implemented more efficiently.

Related issues:

Related to CoCoALib - Feature #628: Complex twin-floats	**New**	**22 Sep 2014**
Related to CoCoALib - Feature #520: Compute inverse in quotient ring (i.e. di...	**Closed**	**04 Apr 2014**
Related to CoCoA-5 - Feature #993: New function: RingQQi()? extension of QQ ...	**In Progress**	**14 Dec 2016**

History

#1 - 23 Sep 2014 11:40 - John Abbott

What should be the internal representation of an element of QQ[i]?

1. a pair of BigRat values (real and imag parts)
2. a triple of BigInt values (real, imag and common denom)
3. a 4-tuple of BigInt values (real & imag of numerator, real & imag of denominator)

Representation (3) is unnormalized, *i.e.* the same value has many different representations; division is very simple; addition and multiplication are more costly than in repr (2).

Repr (2) is the most "specialized"; I would expect it to be faster at run-time than the other two reprs.

Repr (1) is perhaps the most natural, but having two separate denominators is likely to be a disadvantage at run-time (more memory space, and slower computation than repr (1))

#2 - 23 Sep 2014 16:30 - John Abbott

Recalling that every ring (except RingZZ) in CoCoALib has a BaseRing and an "extension type". What should the BaseRing and "extension type" of QQ[i] be?

Ideally it should behave as if it were a quotient of a polynomial extension. But if we do this then it must be possible to obtain a "representative" being a univariate polynomial in the polynomial ring QQ[i].

Perhaps there should be a special "simple algebraic extension" type whose BaseRing is the coeff ring, and the intermediate polynomial ring is not accessible?

#3 - 14 Dec 2016 15:47 - John Abbott

- Related to Feature #993: New function: RingQQi()? extension of QQ with imaginary unit added