CoCoALib - Feature #520

Compute inverse in quotient ring (i.e. division in algebraic extn)

04 Apr 2014 00:37 - John Abbott

Status:	Closed	**Start date:**	04 Apr 2014
Priority:	High	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	New Function	**Estimated time:**	10.00 hours
Target version:	CoCoALib-0.99536 June 2015	**Spent time:**	9.35 hours

Description

Implement "division" in a quotient ring.

Related issues:

Related to CoCoALib - Feature #627: Gaussian integer and rationals ZZi, QQi	**New**	**22 Sep 2014**
Related to CoCoALib - Design #871: Redesign ideals	**New**	**26 Apr 2016**
Related to CoCoALib - Feature #107: Recognizing finite fields	**Closed**	**19 Mar 2012**

History

#1 - 04 Apr 2014 00:41 - John Abbott

A robust general solution is to use GenRepr:
inside R/I
invert element alpha
Check that 1 isin ideal(alpha)+I
if not, there's no inverse
if so, compute GenRepr(1, ideal(alpha, g1, ..., gn))
result is coeff corr to alpha (it's residue class in R/I, of course).

This approach will work even if R/I is not "zero-dimensional".
If R/I is an algebraic field extn, maybe linear algebra would be faster?

#2 - 04 Apr 2014 00:42 - John Abbott

- Category set to New Function

- Estimated time set to 5.00 h

#3 - 04 Apr 2014 15:24 - John Abbott

Anna suggests that elim may be quicker/simpler/better?

#4 - 04 Apr 2014 15:25 - John Abbott

- Target version changed from CoCoALib-0.99533 Easter14 to CoCoALib-0.99534 Seoul14

#5 - 10 Jul 2014 14:23 - John Abbott

- Target version changed from CoCoALib-0.99534 Seoul14 to CoCoALib-1.0

#6 - 10 Apr 2015 10:40 - Anna Maria Bigatti

- Status changed from New to In Progress

- Assignee set to Anna Maria Bigatti

- Priority changed from Normal to High

- % Done changed from 0 to 70

Implemented SparsePolyRingBase::IdealImpl::myDivMod for the 0-dimensional case.

```
/**/ Use R ::= QQ[i];
/**/ QQi := NewQuotientRing(R, ideal(i^2+1));
/**/ use QQi[x];
/**/ 1/i;
(-i)
```

in principle it works.
In practice it needs optimizing and polishing because ideal(this) does not compile (so I used a horrible ideal(myGensValue)) and MultiplicationMatrix might be made more efficient.

#7 - 29 Apr 2015 19:03 - Anna Maria Bigatti

- % Done changed from 70 to 90

- Estimated time changed from 5.00 h to 10.00 h

Polishing up all the code is always long and tedious....
Anyway!
The code is there and it works.
I still have problems with the case

```
/**/ R ::= QQ[i,r];
/**/ K := NewQuotientRing(R, ideal(ReadExpr(R, "i^2+1"),ReadExpr(R, "r^2-1")));
/**/ Use K[x];
/**/ x/i;
```

Slugs:
- when calling num/den there is a call to IsZeroDivisor(den). That's a correct thing to do, but maybe it could be done more efficiently? (for example: try to compute the answer first...)
- when IsZeroDivisor is called and should return false then its ring is not integral! In the case of a QuotientRing one could call myDefiningIdeal->SetPrimeFlag(false)... but how?

#8 - 30 Apr 2015 10:07 - Anna Maria Bigatti

Also x/i is working now.
(will cvs it this afternoon)

#9 - 07 May 2015 13:50 - Anna Maria Bigatti

Anna Maria Bigatti wrote:

> - when IsZeroDivisor is called and should return false then its ring is not integral! In the case of a QuotientRing one could call myDefiningIdeal->SetPrimeFlag(false)... but how?

done: this forced to have the member function myIsZeroDivisor which, for a QuotienRing R/I, may set I's primality flag to false.

#10 - 11 May 2015 14:05 - John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99536 June 2015

#11 - 01 Jul 2015 18:40 - John Abbott

- Status changed from In Progress to Closed

- % Done changed from 90 to 100

No problems arisen in the last month (perhaps not much real testing either?)
Anyway, closing.

#12 - 26 Apr 2016 15:10 - John Abbott

- Related to Design #871: Redesign ideals added

#13 - 27 Jun 2016 08:53 - Anna Maria Bigatti

- Related to Feature #107: Recognizing finite fields added