CoCoALib - Feature #165

FractionField -- only of TrueGCDDomain?

21 May 2012 12:51 - John Abbott

Status:	Closed	**Start date:**	21 May 2012
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Tidying	**Estimated time:**	4.40 hours
Target version:	CoCoALib-0.99534 Seoul14	**Spent time:**	4.00 hours

Description

I've started implementing IsTrueGCDDomain.

I noticed that the ctor for a **FractionField** checks that the base ring is in fact a (true) GCD domain. However other code in PolyRing.C makes an explicit check that the base ring of an already existing FractionField is indeed a TrueGCDDomain.

Technically it is possible to implement a FractionField over a non-GCDDomain (*e.g.* fractions are simply never reduced). Could this ever be useful? [JAA is doubtful] Note that an implementation which never reduced fractions might very easily suffer from exponential growth in the size of its ring elements (unless come from a finite ring).

JAA proposes: the base ring of a **FractionField** must be a TrueGCDDomain (and this must be clearly documented).

If ever it becomes necessary to make a field of fractions over a non-TrueGCDDomain then a new name is used for this new structure, so that existing users of FractionField can rely on the base ring being a TrueGCDDomain.

Related issues:

Related to CoCoA-5 - Support #181: Functions throwing error even though (triv...	**New**	**08 Jun 2012**

History

#1 - 21 May 2012 14:36 - John Abbott

If my proposal is accepted, someone will have to look through the existing code and remove any checks which will have become useless. I have noticed several times a check of the form IsTrueGCDDomain(BaseRing(AsFractionField(..)))

#2 - 21 May 2012 15:09 - Anna Maria Bigatti

John Abbott wrote:

> JAA proposes: the base ring of a **FractionField** must be a TrueGCDDomain (and this must be clearly documented).
>
> If ever it becomes necessary to make a field of fractions over a non-TrueGCDDomain then a new name is used for this new structure, so that existing users of FractionField can rely on the base ring being a TrueGCDDomain.

I agree that, at least for the moment, we only have this case in mind... but I feel uncomfortable not permitting the general case (even though impractical for "big" computations)
...Maybe we could add a member field myBaseRingIsGCDDomain, to highlight that code in case we might decide to convert it into an abstract/concrete class.

#3 - 21 May 2012 17:11 - John Abbott

- Status changed from New to In Progress

After speaking to Anna and Renzo, we concluded that it is better to keep FractionField general.

Anna proposed creating a new function **IsFractionFieldOfGCDDomain** (name still to be finalized) for use in those fns which really want to test that special case.

JAA thinks that it will be tricky to implement clearing denoms of a poly in k[x] where k is a fraction field over a non-gcd ring.

#4 - 21 May 2012 17:12 - John Abbott

- % Done changed from 0 to 70

#5 - 22 May 2012 12:06 - John Abbott

I've just looked at the code for FractionFieldImpl. I must make a separate impl for arithmetic if GCD is not available. The structure will probably be: common "base class", and two concrete derived classes (one "with GCD" and the other "without GCD"). The pseudo-ctor will have to choose the right ctor.

AMB reply:
yes, that's what I had implied with an earlier comment.
What I suggest is that
1 - make a function **IsFractionFieldOfGCDDomain** for simplifying the checks
2 - keep the current restriction that a **FractionField** can only be created on GCDDomain (the only useful and practical ones)
3 - Keep in mind that we could have a general implementation (which requires separating abstract/concrete classes) which is probably useless.

#6 - 01 Apr 2014 17:34 - Anna Maria Bigatti

- Target version set to CoCoALib-0.99533 Easter14

#7 - 04 Apr 2014 17:49 - John Abbott

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

#8 - 09 Jul 2014 18:11 - John Abbott

- Category set to Tidying

Can this be closed in the near future? (i.e. before Seoul)

#9 - 09 Jul 2014 18:19 - Anna Maria Bigatti

I think we'd better be conservative, i.e. only for true GCDRings.
If someone ever asks for the general case then we can activate it (and see whether it is actually usable).

#10 - 11 Jul 2014 12:05 - John Abbott

- Status changed from In Progress to Closed

- Assignee set to John Abbott

- % Done changed from 70 to 100

- Estimated time set to 4.40 h

Here's the conclusion:

- currently FrF only of true GCD domain allowed
- the fn **IsFractionFieldOfGCDDomain** has been impl

A possible future generalization to arbitrary integral domains (incl. fields?) would be via a new concrete class for that special case. I repeat that it would probably be impractical for all but toy computations. Despite comments earlier in this issue, there is apparently already a split into absract base + concrete impl.