CoCoALib - Feature #259

Squarefree(?) GCD-free basis

09 Oct 2012 15:55 - John Abbott

Status:	Closed	**Start date:**	09 Oct 2012
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	New Function	**Estimated time:**	10.70 hours
Target version:	CoCoALib-0.99700	**Spent time:**	10.70 hours

Description

Hensel lifting for univariate GCDs requires a squarefree GCDfree basis.
There is an implementation in GCDfreeBasis.cpkg5; convert this to C++.

Needs a good GCD impl to work properly -- sounds like a circular argument!

Related issues:

Related to CoCoA-5 - Support #242: CoCoA-5 Projects for students (e.g. credit...	**In Progress**	**28 Sep 2012**
Related to CoCoALib - Feature #4: Squarefree GCD-free basis	**Rejected**	**19 Oct 2011**
Related to CoCoALib - Bug #154: GCD normalization (e.g. monic)	**In Progress**	**07 May 2012**
Related to CoCoA-5 - Support #1240: John's visit Feb 2019	**Closed**	**08 Feb 2019**

History

#1 - 01 Aug 2014 08:59 - Anna Maria Bigatti

- Target version set to CoCoALib-1.0

#2 - 24 Nov 2016 13:19 - John Abbott

- Category set to New Function

- Status changed from New to In Progress

- % Done changed from 0 to 10

I have written a first version (for RingElem) by translating almost directly the impl in GCDFreeBasis.cpkg5. I have not yet tested it, nor even checked it in. It does just the GCDfree part, not the squarefree part.

Note that the CoCoA-5 impl was just for integers; the new one is for any ring elem (in a true GCD domain).

#3 - 24 Nov 2016 13:57 - John Abbott

- Related to Bug #154: GCD normalization (e.g. monic) added

#4 - 30 Jun 2017 13:18 - John Abbott

Mostly a wake-up call. There is already an implementation in GCDFreeBasis.C.
Decide which data-structures to use (principally vector<RingElem> or some new type).

The "refine" function could actually be a member function.

#5 - 21 Jun 2018 22:39 - John Abbott

- Status changed from In Progress to Resolved

- % Done changed from 10 to 60

I have impls of GCDFreeBasis for RingElem and for BigInt.
Not yet checked in/ No tests; one simple example.

#6 - 25 Jun 2018 15:25 - John Abbott

- Assignee set to John Abbott

- % Done changed from 60 to 70

I have checked in the code. There is doc, but no tests.

I am not happy with the class names: **GCDFreeBasis_BigInt** and **GCDFreeBasis_RingElem**.
Since they are classes the names have to be different (or I could use templates -- awkward in this case).

Also I am slightly unhappy about the root of the name **GCDFreeBasis**. In this case "GCDFree" means "coprime", so why not say "coprime"? Also I believe that there is an expression "factor base" rather than "factor basis". So a better root name might be **CoprimeFactorBase**. In a sense it is nice to have the substring "Factor" in the name.

Opinions? Ideas? Suggestions?

#7 - 03 Aug 2018 17:10 - John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99650 November 2019

#8 - 26 Feb 2019 17:21 - John Abbott

Should **GCDFreeBasis_BigInt** and **GCDFreeBasis_RingElem** be renamed to **CoprimeFactorBasis_BigInt** and **CoprimeFactorBasis_RingElem**?

#9 - 26 Feb 2019 17:21 - John Abbott

- Related to Support #1240: John's visit Feb 2019 added

#10 - 01 Oct 2019 11:37 - John Abbott

- Status changed from Resolved to Feedback

- Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-0.99700

- % Done changed from 70 to 90

I changed the names (not sure when).

A problem with the integer version is that the "squarefree" part is potentially costly to achieve -- I believe it requires factorization.

At the moment I am tempted to skip the "squarefree" guarantee, and say that it is the caller's responsibility.
Anyway is it better to do squarefree factorization and then CoprimeFactorBasis, or *vice versa?*
Probably there are situations where one approach is better, and situations where the other is better...

Moved to "feedback".

#11 - 09 Jan 2020 22:22 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time set to 10.70 h

These fns were already mentioned in the previous release (0.99650).
Closing after spending 3 months in *feedback*.