CoCoALib - Feature #1197

IsZeroDet: new fn

26 Jun 2018 14:40 - John Abbott

Status:	In Progress	**Start date:**	26 Jun 2018
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	40%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	4.75 hours

Description

Implement IsZeroDet which should allow a fast modular impl.

Related issues:

Related to CoCoALib - Slug #1057: Slug: Polynomial ring contructor slow with ...	**In Progress**	**04 May 2017**
Related to CoCoALib - Bug #1641: gcd does not recognize univariate input	**Closed**	**20 Dec 2021**

History

#1 - 05 Apr 2019 15:46 - John Abbott

The idea is simple: compute the det modulo various primes, and if any is non-zero then we know the non-modular det is non-zero.

Probably this fn should accept a VerificationLevel parameter. It might even be nice if the "verification level" is interpreted as the log (base 2?) of the product of the moduluses tried -- this would make it more-or-less independent of the size of the primes we choose as the moduli.

The idea extends trivially to matrices over QQ (without having to clear denoms).

#2 - 09 Apr 2019 10:55 - John Abbott

- Related to Slug #1057: Slug: Polynomial ring contructor slow with (big) matrix ordering added

#3 - 09 Apr 2019 10:57 - John Abbott

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

#4 - 08 Jan 2020 22:58 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

#5 - 27 Feb 2020 11:50 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

I have just made a very simplistic first impl: it simply computes the det, and checks whether it is zero. (also checked-in)

This first impl does not accept a VerificationLevel optional arg.

The fn is accessible from CoCoA-5; it may be useful for the school in Vietnam...

STILL TO DO: make a proper implementation & consider adding in a VerificationLevel parameter.

Special cases:

- over finite field -- no short-cut comes to mind
- over QQ or ZZ we can use "early termination" in a CRT approach; the existing code should be modified slightly
- not sure what to do about small dimension mats over QQ or ZZ (currently the det is computed by special purpose code)
- over a poly ring, we could make some random subst for the indets; if result has nz det then orig mat has nz det

#6 - 06 Oct 2020 14:50 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

#7 - 21 Jan 2022 12:36 - John Abbott

- Related to Bug #1641: gcd does not recognize univariate input added

#8 - 31 Jan 2024 22:26 - John Abbott

- Assignee set to John Abbott

- % Done changed from 10 to 40

There is some non-trivial code in **MatrixOps-IsZeroDet.C** developed from a project in Passau

```
With contributions from Mohanad Baraghith (Uni Passau)
```

#9 - 08 Mar 2024 17:58 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880