CoCoALib - Design #1642

interreduce: make monic if over finite field?

21 Dec 2021 21:56 - John Abbott

Status:	In Progress	**Start date:**	21 Dec 2021
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	20%
Category:	Improving	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	0.55 hour

Description

While working on a Sudoku exercise I noticed that interreduced sometimes return non-monic polynomials: coeff field is ZZ/(19).
Does it make sense to make the polys monic?

Discuss; maybe implement!

Related issues:

Related to CoCoALib - Feature #1488: BuiltIn Interreduce-Function	**Closed**	**15 Sep 2020**

History

#1 - 21 Dec 2021 21:56 - John Abbott

- Related to Feature #1488: BuiltIn Interreduce-Function added

#2 - 16 Feb 2024 09:42 - John Abbott

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

Would we also want to have interreduced work when the coeff ring is ZZ?
If so, we need to think about the semantics. If not, the code should give an error if the coeffs are not a field?

#3 - 16 Feb 2024 10:19 - Anna Maria Bigatti

John Abbott wrote:

> Would we also want to have interreduced work when the coeff ring is ZZ?

Not until someone asks for it (and tells us the expected semantics)

#4 - 16 Feb 2024 17:02 - John Abbott

The source code seems to be in **SparsePolyOps-vector.C** around line 32.

#5 - 16 Feb 2024 20:44 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 20

I have made a first impl. Anna is sceptical, partly because the code behaves differently depending on the coefficient field (so computation modulo p is not simply reduction, but rescaled). She also thought that there would be no performance gain, and perhaps indeed a penalty from the cost of rescaling polynomials. She might be right...

Unhelpfully, I did not give an explicit test case. Maybe I can invent one?