CoCoALib - Feature #1030

IsInRadical: case of homog ideal

14 Mar 2017 13:54 - John Abbott

Status:	Closed	**Start date:**	14 Mar 2017
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Improving	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99560	**Spent time:**	10.45 hours

Description

Currently IsInRadical is defined in a CPKG5, but may soon be translated to CoCoALib.

The case of a homog poly in a homog ideal is handled specially.

I wonder if we cannot improve it by simply testing each homog component of the poly for membership in the radical.

Related issues:

Related to CoCoA-5 - Bug #1032: IsInRadical: fragile code	**Closed**	**17 Mar 2017**
Related to CoCoALib - Feature #1033: Split poly into homog parts	**Closed**	**17 Mar 2017**
Related to CoCoA-5 - Bug #1610: IsInRadical: some more little bugs	**Closed**	**27 Sep 2021**

History

#1 - 14 Mar 2017 14:00 - John Abbott

- Subject changed from IsInRadical: case of homog poly to IsInRadical: case of homog ideal

- Status changed from New to In Progress

- % Done changed from 0 to 10

I think that if the ideal I is homog then IsInRadical(f,I) is the same as the logical-and of IsInRadical(f_d,I) for all homog components f_d of f. I do not know whether it is faster to compute it this way... or maybe my maths is wrong?

#2 - 14 Mar 2017 14:18 - Anna Maria Bigatti

John Abbott wrote:

> I think that if the ideal I is homog then IsInRadical(f,I) is the same as the logical-and of IsInRadical(f_d,I) for all homog components f_d of f. I do not know whether it is faster to compute it this way... or maybe my maths is wrong?

correct: f = f_d + ... (f_d homog of deg d = deg(f)).
f^n = (f_d)^n + ... in I implies (f_d)^n in I implies f_d is in radical(I)

#3 - 17 Mar 2017 11:07 - John Abbott

- Related to Bug #1032: IsInRadical: fragile code added

#4 - 17 Mar 2017 11:17 - John Abbott

- Related to Feature #1033: Split poly into homog parts added

#5 - 19 Jul 2017 19:14 - John Abbott

- Status changed from In Progress to Feedback

- Assignee set to John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99560

- % Done changed from 10 to 90

The CoCoA-5 package was translated into C++ by some students at Kassel.
I have cleaned up the resulting code, and checked it in: see files **RadicalMembership**
I have added doc and a test (but no example).
I have made the fns available via CoCoA-5; the old package is still there, but I have changed the fn names to avoid clashes. Probably the package should simply be deleted (perhaps after a bit more testing?)

I have added a couple of "heuristic tricks" to **IsInRadical**, as otherwise computation times can be very long (esp, when the polynomial is not in the radical). The trick is just to see if a generator (or RGB element) is not square-free; if so, add as new generator the "radical" of that generator.

Note that SqFreeFactor can be slow when coeffs are in a finite field (since GCD is still via a GBasis computation); so the trick is not applied to "big" polys.

#6 - 22 Jul 2017 15:18 - John Abbott

I wonder if either of the following ideas could be completed into an algorithm (with reliable output):

1. if the ideal is not 0-dim, adjoin some random linear polys (or linear forms) to the gens possibly making the ideal 0-dim, then test for radical membership. If poly is not in radical of extended ideal, it is surely not in the radical of the original ideal; the other case is less clear.
2. if ideal is over QQ, try a modular approach; perhaps use MinPowerInIdeal to predict power of original power which ought to be in the original ideal (and then test that power directly).

Is it true that every (polynomial) ideal *I* has an "exponent" *exp(I)* such that for any polynomial IsInRadical(f,I) iff f^exp(I) IsIn I. I'm not sure how the exponent could be computed.

#7 - 10 Nov 2017 12:23 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

#8 - 28 Nov 2017 17:56 - John Abbott

- Description updated

#9 - 15 Feb 2024 16:13 - Anna Maria Bigatti

- Related to Bug #1610: IsInRadical: some more little bugs added