

CoCoA-5 - Bug #1032

IsInRadical: fragile code

17 Mar 2017 11:07 - John Abbott

Status:	Closed	Start date:	17 Mar 2017
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	enhancing/improving	Estimated time:	1.11 hour
Target version:	CoCoA-5.2.2	Spent time:	1.20 hour
Description			
The current impls of IsInRadical and MinPowerInRadical are fragile: they give errors when they should/need not.			
Related issues:			
Related to CoCoALib - Feature #1030: IsInRadical: case of homog ideal		Closed	14 Mar 2017
Related to CoCoA-5 - Slug #1141: IsInRadical: for monomial ideals		New	15 Dec 2017
Related to CoCoA-5 - Bug #1610: IsInRadical: some more little bugs		Closed	27 Sep 2021

History

#1 - 17 Mar 2017 11:07 - John Abbott

- Related to Feature #1030: IsInRadical: case of homog ideal added

#2 - 17 Mar 2017 11:11 - John Abbott

Here are some cases where behaviour is not ideal:

```
use QQ[x,y,z];
TmpI := ideal(x+y);
J := elim(x,I); // ideal without generators
IsInRadical(x,J); --> ERROR: empty list or vector
IsInRadical(zero(R),J); --> ERROR: empty list or vector

I0 := ideal(zero(R));
IsInRadical(zero(R),I0); // --> non-zero ringelem required

I1 := ideal(one(R));
IsInRadical(0,I1); // --> first arg must be POLY or IDEAL
IsInRadical(1,I1); // --> first arg must be POLY or IDEAL
IsInRadical(1/2,I1); // --> first arg must be POLY or IDEAL
```

Since **IsIn** accepts the combination RAT IsIn IDEAL, then it seems reasonable that IsInRadical should also accept INT or RAT as the type for the element.

### #3 - 17 Mar 2017 11:13 - John Abbott

Here are some examples which suggest that it may be better to decompose a poly into its homog parts when testing membership in the radical of a homog ideal:

```
Ihomog := ideal(x^999);
[IsInRadical(x^k,Ihomog) | k in 1..20];
IsInRadical(x^2+x,Ihomog); // SLUG, much slower than prev line
IsInRadical(x*sum([random(-99,99)*x^k | k in 0..10]), Ihomog); // SLUG!!!

Ihomog2 := ideal(x^999,x^1000+x^999);
[IsInRadical(x^k,Ihomog2) | k in 1..20];
IsInRadical(x^2+x,Ihomog2); // SLUG, much slower than prev line
IsInRadical(x*sum([random(-99,99)*x^k | k in 0..10]), Ihomog2); // SLUG!!!

IsInRadical(J, I0); // --> OK
IsInRadical(J, I1); // --> OK

[MinPowerInIdeal(x^k, Ihomog) | k in 1..20];
MinPowerInIdeal(x^2+x, Ihomog);
MinPowerInIdeal(x*(x^10-1)/(x-1), Ihomog);

Ihomogxy := ideal(x^400,y^400);
MinPowerInIdeal((x^10-y^10)/(x-y), Ihomogxy); --> about 20s
S := support((x^10-y^10)/(x-y));
[MinPowerInIdeal(t, Ihomogxy) | t in S]; --> instant
f := x*(x^4-1)/(x-1)*y*(y^4-1)/(y-1);
S := support(f);
[MinPowerInIdeal(t, Ihomogxy) | t in S]; --> < 1 s
MinPowerInIdeal(f, Ihomogxy); --> > 200s
Ihomogxy2 := ideal(x^400+y^401,y^400);
MinPowerInIdeal((x^10-y^10)/(x-y), Ihomogxy2);
//MinPowerInIdeal(x*(x^4-1)/(x-1)*y*(y^4-1)/(y-1), Ihomogxy2);
```

### #4 - 28 Nov 2017 14:04 - John Abbott

- Status changed from New to Resolved
- Assignee set to John Abbott
- Target version changed from CoCoA-5.?.? to CoCoA-5.2.2

- % Done changed from 0 to 80

- Estimated time set to 1.11 h

Most of these problems have been resolved by porting the code to CoCoALib (thanks to Alice Moallemy, Marvin Brandenstein, Carsten Dettmar).

There remain 2 slugs:

```
Ihomogxy := ideal(x^400,y^400);
f := x*((x^4-1)/(x-1))*y*((y^4-1)/(y-1));
S := support(f);
[MinPowerInIdeal(t, Ihomogxy) | t in S]; --> less than 1 s
MinPowerInIdeal(f, Ihomogxy); --> about 100s
Ihomogxy2 := ideal(x^400+y^401,y^400);
MinPowerInIdeal(f, Ihomogxy2); --> about 140s
```

#### #5 - 28 Nov 2017 14:56 - Anna Maria Bigatti

My guess (I do not know the code) is that there is something optimized for monomial ideals...

Anyway, just looking at the code, I think that `RadicalHelpers(const vector<RingElem>& G)` could use `radical(G[i])`, instead of calling `SqFreeFactor(G[i])`; etc

#### #6 - 15 Dec 2017 15:37 - John Abbott

- Related to Slug #1141: *IsInRadical*: for monomial ideals added

#### #7 - 15 Dec 2017 15:37 - John Abbott

- Status changed from *Resolved* to *Closed*

- % Done changed from 80 to 100

#### #8 - 27 Sep 2021 09:53 - Anna Maria Bigatti

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