

## CoCoALib - Feature #796

### CoCoALib function for radical (or SqFree) of a polynomial

05 Nov 2015 16:45 - Anna Maria Bigatti

<b>Status:</b>	Closed	<b>Start date:</b>	05 Nov 2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	New Function	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CoCoALib-0.99560	<b>Spent time:</b>	8.25 hours
<b>Description</b>			
This problem is generally easier than SqFreeFactor, indeed in SqFreeFactorPosDerChar0 it is just computed by the first 3 lines. How easy is it to extract this function out of all cases considerend in SqFreeFactor?			
<b>Related issues:</b>			
Related to CoCoALib - Feature #39: Squarefree factorization		<b>Closed</b>	<b>30 Nov 2011</b>
Related to CoCoALib - Feature #45: Squarefree factorization - univariate pol...		<b>Closed</b>	<b>30 Nov 2011</b>
Related to CoCoALib - Feature #46: Squarefree factorization - univariate pol...		<b>Closed</b>	<b>20 Dec 2011</b>
Related to CoCoALib - Feature #47: Squarefree factorization - multivariate po...		<b>Closed</b>	<b>30 Nov 2011</b>
Related to CoCoALib - Feature #947: IsRadical for ideal?		<b>In Progress</b>	<b>18 Oct 2016</b>
Related to CoCoALib - Design #950: factor and SmoothFactor for integers --> F...		<b>Closed</b>	<b>20 Oct 2016</b>
Related to CoCoALib - Feature #951: New function: IsSqFree		<b>Closed</b>	<b>24 Oct 2016</b>

### History

#### #1 - 18 Oct 2016 14:56 - John Abbott

- Related to Feature #947: IsRadical for ideal? added

#### #2 - 18 Oct 2016 15:12 - Anna Maria Bigatti

This would be quite useful. For the time being should we just add

```
define SqFree(f)
  return product(SqFreeFactor(f).factors);
enddefine;
```

?

#### #3 - 20 Oct 2016 13:30 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

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

- % Done changed from 10 to 50

I have an implementation in CoCoALib. It took so long because of a "mysterious bug" in ContentFreeFactor.

My current implementation works only for RingZZ and a polynomial ring over QQ or over a small finite field.

Curiously it seems to be faster for polynomials over QQ than over a finite field: in the former case it can use the simple formula  $f/\gcd(f,f')$  whereas in the latter it computes a squarefree factorization then multiplies the factors together.

The case of monomials is handled specially.

**NOTE** Other cases could be handled: e.g. polynomials with integer coeffs.

#### #4 - 20 Oct 2016 15:44 - John Abbott

Before checking in which name should I use?

Currently I have used **radical** in CoCoALib, and **rad** in CoCoA-5 (to avoid a clash with the fn radical defined in radical.cpkg).

Suggestions?

#### #5 - 20 Oct 2016 16:10 - Anna Maria Bigatti

John Abbott wrote:

Currently I have used **radical** in CoCoALib, and **rad** in CoCoA-5 (to avoid a clash with the fn radical defined in radical.cpkg).

we could indeed use rad(l), rad(f),... which seems quite standard in all environments.

#### #6 - 20 Oct 2016 20:08 - John Abbott

- Status changed from *In Progress* to *Resolved*

- % Done changed from 50 to 60

Checked in the code. Also two new tests test-NumTheory6 and test-SparsePolyRing3.  
No documentation yet.

Names not yet decided. JAA finds **radical** clearer than **rad** (but it is also longer).

#### #7 - 20 Oct 2016 20:10 - John Abbott

- Related to Design #950: factor and SmoothFactor for integers --> FactorINT, FactorINT\_TrialDiv, FactorINT\_PollardRho added

#### #8 - 24 Oct 2016 16:54 - Anna Maria Bigatti

After spending some time thinking and writing, I realized that we would need a IsSqFree(f) which could be considerably faster than computing and checking f=radical(f). (especially when it isn't)

#### #9 - 24 Oct 2016 17:14 - John Abbott

I like the idea of a fn IsSqFree; it certainly could be faster than testing equality with the radical (and also clearer?)

I find that **IsSqFree** is easier to understand than **IsRadical** for ringelems (and perhaps BigInt).  
But then it is inconsistent to have IsSqFree as the test, and radical to compute the "maximal square-free factor".

Unfortunately, I also find SqFree to be a poor choice for the name of the fn which computes the "maximal square-free factor".

Not sure how to resolve this naming problem.

**#10 - 24 Oct 2016 17:31 - Anna Maria Bigatti**

- *Related to Feature #951: New function: IsSqFree added*

**#11 - 06 Nov 2017 14:24 - John Abbott**

- *% Done changed from 60 to 80*

It seems that this issue is practically finished except for the question of the name of the function. The main choice seems to be between: **rad**, **radical** or **SqFree**.

Let's decide; then we can close the issue.

**#12 - 06 Nov 2017 15:20 - Anna Maria Bigatti**

John Abbott wrote:

It seems that this issue is practically finished except for the question of the name of the function. The main choice seems to be between: **rad**, **radical** or **SqFree**.

I vote for **radical**: I cannot see a similar function in Macaulay2; however in Singular it is called sqrfree and its result is by default the square free factorization (and with flag "3" the radical of the polynomial).

**#13 - 06 Nov 2017 15:31 - John Abbott**

- *Status changed from Resolved to Closed*

- *% Done changed from 80 to 100*

OK, let's use **radical** then (both for ideals and ring elements); it is the clearest.

Closing.