

## CoCoALib - Feature #899

### IsMaximal, IsPrimary for IDEAL (in cocoalib)

27 Jun 2016 08:50 - Anna Maria Bigatti

<b>Status:</b>	Closed	<b>Start date:</b>	27 Jun 2016
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	New Function	<b>Estimated time:</b>	6.00 hours
<b>Target version:</b>	CoCoALib-0.99550 spring 2017	<b>Spent time:</b>	5.75 hours
<b>Description</b>			
2017-11: actual setting of flags is discussed in <a href="#">#924</a>			
<b>Related issues:</b>			
Related to CoCoALib - Feature #107: Recognizing finite fields		<b>Closed</b>	<b>19 Mar 2012</b>
Related to CoCoA-5 - Feature #844: IsMaximal, IsPrimary for IDEAL (in cocoa l...		<b>Closed</b>	<b>16 Feb 2016</b>
Related to CoCoALib - Feature #361: implement IsPrime3, IsMaximal3		<b>Closed</b>	<b>29 May 2013</b>
Related to CoCoALib - Design #513: Quick/correct flag for bool3 fns		<b>In Progress</b>	<b>02 Apr 2014</b>
Related to CoCoALib - Design #924: FlagManager for bool/bool3 flags		<b>New</b>	<b>19 Sep 2016</b>
Related to CoCoA-5 - Bug #870: GBasis of product of ideals is wrong (Vadim Tr...		<b>Closed</b>	<b>26 Apr 2016</b>
Related to CoCoALib - Feature #1178: New function: myPrimaryDecomposition_0dim		<b>Closed</b>	<b>06 Apr 2018</b>
Related to CoCoALib - Design #1409: myTestIsPrimary & Co. : fix design		<b>Closed</b>	<b>03 Feb 2020</b>

#### History

##### #1 - 27 Jun 2016 08:51 - Anna Maria Bigatti

- Related to Feature #107: Recognizing finite fields added

##### #2 - 27 Jun 2016 08:55 - Anna Maria Bigatti

- % Done changed from 0 to 60

Implemented, need proper testing.

##### #3 - 27 Jun 2016 08:58 - Anna Maria Bigatti

- Related to Feature #844: IsMaximal, IsPrimary for IDEAL (in cocoa language) added

##### #4 - 02 Aug 2016 14:21 - Anna Maria Bigatti

- Related to Feature #361: implement IsPrime3, IsMaximal3 added

##### #5 - 07 Sep 2016 16:49 - Anna Maria Bigatti

After working on it, I realized that usually the function actually doing the "test" computation is also the function knowing whether other flags may be set. For example the test function for IsPrime in  $K[x]$  can also set the IsMaximal flag. So now I believe that the test function should set the flag and all the other flags it knows of. In other words: **the flag should be set by the test function.**

##### #6 - 08 Sep 2016 09:17 - John Abbott

- Status changed from New to In Progress

After speaking to Anna about this yesterday...

It seems best that the flags be assigned to only via specific functions (and not by direct C++ assignment to the flag); these specific functions would then embody any relationship between the flags. For instance, the specific fn for the "maximal" flag, when setting it to true3 would also set the "prime" flag to true3.

These flag-assigning fns can be called by any fn which is able to deduce the correct value (or by an explicit user call). Presumably an error must be signalled if there is an attempt to change the flag from true3 to false3 or vice versa.

What names should these assigning fns have? Anna suggested something like myAssignMaximalFlag.

## #7 - 08 Sep 2016 09:30 - John Abbott

I'm not sure if this suggestion is a good one or a bad one.

If the flags are simply data members of type bool3 then the compiler offers no protection against using C++ assignment to these fields (rather than calling the specific flag-assigning fns).

We could get help from the compiler by putting all the flags in a separate, internal class; this class then offers public fns for assigning the flags, and for getting their values. Here is an outline of what I mean:

```
class ideal
{
    class FlagManager
    {
    private: // data members
        bool3 myMaximalFlag;
        bool3 myPrimeFlag;
    public:
        FlagManager() {} // default ctor OK???
        void myAssignMaximalFlag(bool value) { .... };
        void myAssignPrimeFlag(bool value) { .... };
        bool3 myGetMaximalFlag() const { return myMaximalFlag; };
        bool3 myGetPrimeFlag() const { ... };
    }
    FlagManager myFlags;
};
```

I am not proposing the names I have used here -- I think they are too cumbersome: for instance, a mem fn of ideal which has proved the ideal to be maximal would have to call myFlags.myAssignMaximalFlag(true) which says Flag twice...

Is it better to have a single setter fn which accepts a bool as arg, or have two setter fns (one with implicit arg true, the other with implicit arg false)? JAA currently prefers a single setter fn with a bool arg, as it "feels" a bit more flexible (even though no concrete example springs to mind).

**#8 - 08 Sep 2016 11:16 - Anna Maria Bigatti**

I have a question: I have the function `IsRadicalSPR(const ideal& I)` which, in the way of determining if `I` is radical, may detect maximality. This is not a member function: should it be? it is effectively an auxiliary function for `IsRadical`.

**#9 - 08 Sep 2016 11:19 - Anna Maria Bigatti**

- Related to Feature #873: `bool3`: add some logical operations? added

**#10 - 08 Sep 2016 11:19 - Anna Maria Bigatti**

- Related to deleted (Feature #873: `bool3`: add some logical operations?)

**#11 - 08 Sep 2016 11:21 - Anna Maria Bigatti**

- Related to Design #513: Quick/correct flag for `bool3` fns added

**#12 - 08 Sep 2016 14:05 - Anna Maria Bigatti**

John Abbott wrote:

If the flags are simply data members of type `bool3` then the compiler offers no protection against using C++ assignment to these fields (rather than calling the specific flag-assigning fns).

We could get help from the compiler by putting all the flags in a separate, internal class; this class then offers public fns for assigning the flags, and for getting their values. Here is an outline of what I mean:

I really like this idea, both for the safety, and also for the cleanliness: ideals have lots of 3flags...  
I'll try doing something along those lines.

**#13 - 16 Sep 2016 08:31 - Anna Maria Bigatti**

- Related to Design #921: Design `GBMill` added

**#14 - 19 Sep 2016 07:49 - Anna Maria Bigatti**

- Related to Design #924: `FlagManager` for `bool/bool3` flags added

**#15 - 19 Sep 2016 07:50 - Anna Maria Bigatti**

- Related to deleted (Design #921: Design `GBMill`)

**#16 - 21 Sep 2016 07:36 - Anna Maria Bigatti**

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

- Priority changed from *Normal* to *High*

- % Done changed from 60 to 100

- Estimated time set to 6.00 h

The function themselves work fine. The discussion now is about the implementation of the `bool3` flags, which is a more general question, so it is moved to [#924](#).

**#17 - 01 Mar 2017 14:17 - Anna Maria Bigatti**

- Related to Bug #870: GBasis of product of ideals is wrong (Vadim Tropashko) --> `I.myReset()` added

**#18 - 25 Nov 2017 10:40 - Anna Maria Bigatti**

- Description updated

**#19 - 06 Apr 2018 20:31 - Anna Maria Bigatti**

- Related to Feature #1178: New function: `myPrimaryDecomposition_0dim` added

**#20 - 03 Feb 2020 12:59 - Anna Maria Bigatti**

- Related to Design #1409: `myTestIsPrimary` & Co. : fix design added