CoCoALib - Design #1326

Modify function myElim so that it returns ideal? (not quite)

03 Oct 2019 17:18 - Anna Maria Bigatti

Status:	Closed	**Start date:**	03 Oct 2019
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	Tidying	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99850	**Spent time:**	4.95 hours

Description

Currently we have

```
void SparsePolyRingBase::IdealImpl::myElim(const std::vector<RingElem>& ElimIndets)
```

should we modify it so that the ideal is not modified and returns the elimination

```
ideal SparsePolyRingBase::IdealImpl::myElim(const std::vector<RingElem>& ElimIndets) const
```

?

The current design of several functions on ideals modify **this**, but I think it is unnatural for the general cocoalib design.

2024-03 modified like this: J->myAssignElim(I, ElimIndets)

Related issues:

Related to CoCoALib - Feature #813: Implement "elim" in CoCoALib	**Feedback**	**23 Nov 2015**
Related to CoCoALib - Slug #777: SLUG: elimination	**In Progress**	**15 Sep 2015**
Related to CoCoALib - Feature #1619: Make saturate available in CoCoALib	**Closed**	**15 Oct 2021**
Related to CoCoA-5 - Bug #1560: elim for modules	**New**	**11 Jan 2021**
Related to CoCoALib - Design #1767: Finalize design for ideals in CoCoALib	**In Progress**	**31 Oct 2023**

History

#1 - 03 Oct 2019 17:18 - Anna Maria Bigatti

- Related to Feature #813: Implement "elim" in CoCoALib added

#2 - 03 Oct 2019 17:19 - Anna Maria Bigatti

- Related to Slug #777: SLUG: elimination added

#3 - 12 Feb 2020 16:11 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

I prefer to create a new ideal, and not to change the existing one.

Note that an exception-safe impl almost certainly has to create a new ideal for the result anyway...

#4 - 03 Nov 2021 17:00 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

#5 - 21 Jan 2022 12:24 - John Abbott

- Related to Feature #1619: Make saturate available in CoCoALib added

#6 - 14 Mar 2024 10:02 - Anna Maria Bigatti

- % Done changed from 0 to 20

It seems we agree on changing this interface, so I consider this approved.
I'll go on with this, so we can proceed also with the related issues.

#7 - 14 Mar 2024 17:58 - Anna Maria Bigatti

I'm halfway through the process, but this is tricky.
[I think this is interesting, because it's the first try in changing the design for ideals]

myElim is a member function of the class IdealBase, so returning an ideal (smart pointer) doesn't feel quite right.

I would like to swap the computed generators (vector of RingElem) into the new ideal, but to do that I'd need to call ourGetPtr (dynamic_cast), and also that does not seem right.

#8 - 15 Mar 2024 14:21 - Anna Maria Bigatti

- Status changed from New to Resolved

- % Done changed from 20 to 70

As explained in #note-7, I didn't like the first approach.
So I changed the internal functions as J->myAssignElim(I, ElimIndets), and the code came up more nicely.
Not 100% convinced, but surely better than as I->myElim(ElimIndets) in terms of meaning.

Then I implemented elim proper, and called it instead of the old myElim.
Nice.

#9 - 15 Mar 2024 14:22 - Anna Maria Bigatti

- Description updated

#10 - 15 Mar 2024 14:24 - Anna Maria Bigatti

New question: elim(I, X) (currently in CoCoALib) or elim(X, I) (traditionally in CoCoA)?

They have to be same!!
Investigate what other systems do, and decide.

#11 - 15 Mar 2024 14:44 - Anna Maria Bigatti

Anna Maria Bigatti wrote:

> New question: elim(I, X) (currently in CoCoALib) or elim(X, I) (traditionally in CoCoA)?
>
> They have to be same!!
> Investigate what other systems do, and decide.

```
Singular: elim(i,3..4);
Oscar: eliminate(I::PBWAlgIdeal, V::Vector{<:PBWAlgElem}; ordering = nothing)
M2: eliminate(x,ideal(f,g))
Magma: EliminationIdeal(I, k: parameters)
```

#12 - 15 Mar 2024 14:54 - Anna Maria Bigatti

elim of (I,X) or (X,I)?

Pro for (I,X)

1. respects the rule "more structured argument first"
2. like Singular/Oscar and Magma

Pro for (X,I)

1. respects the rule "as you would say in words": "eliminate X from I"
2. respects backward compatibility in CoCoA
3. like Macaulay/2

In fact, there would be no ambiguity in providing both ways (a bit tedious, but possible), but I don't think it is a good idea.

#13 - 15 Mar 2024 15:02 - Anna Maria Bigatti

Anna Maria Bigatti wrote:

> elim of (I,X) or (X,I)?

comparison with other functions in CoCoA/CoCoALib: (I,X) seems winning (but breaks CoCoA backward compatibility)

```
deg(x*y^2+y, x)
homog(x^3-y, w)
CoeffListWRT(F, y)
```

#14 - 15 Mar 2024 15:04 - Anna Maria Bigatti

- Related to Bug #1560: elim for modules added

#15 - 15 Mar 2024 16:50 - Anna Maria Bigatti

- Related to Design #1767: Finalize design for ideals in CoCoALib added

#16 - 18 Mar 2024 16:02 - Anna Maria Bigatti

- Subject changed from Modify function myElim so that it returns ideal to Modify function myElim so that it returns ideal? (not quite)

- Description updated

#17 - 18 Mar 2024 16:03 - Anna Maria Bigatti

- Status changed from Resolved to Closed

- % Done changed from 70 to 100