

## CoCoALib - Feature #1763

### implement ideal(R) for zero ideal, with no generators?

07 Sep 2023 08:31 - Anna Maria Bigatti

<b>Status:</b>	Rejected	<b>Start date:</b>	07 Sep 2023
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	New Function	<b>Estimated time:</b>	1.01 hour
<b>Target version:</b>	CoCoALib-0.99850	<b>Spent time:</b>	0.90 hour
<b>Description</b>			
I'm fairly sure we used to have the constructor ideal(R) for an ideal with no generators, but I cannot find any trace of it (and/or why we do not have it). Of course we can write explicitly ideal(R, v) with an empty vector, but that's ugly.  Should I add the constructor ideal(R)?  <b>2024-02</b> No longer needed: just call ideal(zero(R))			
<b>Related issues:</b>			
Related to CoCoALib - Design #1647: Suppress zero from ideal generators? Det...		<b>Closed</b>	<b>20 Jan 2022</b>

#### History

##### #1 - 07 Sep 2023 11:37 - John Abbott

Mmm, don't remember... I'm still on holiday (sort of).

We can do **ideal(zero(P))** and **ideal(P,EmptyList)**.

Maybe the argument was that ideal(P) is less clear to read than either of the alternatives in the line above?

Do we have good "use cases" where someone really want to create a zero ideal with no generators rather than than a zero ideal containing one zero generator?

We could also consider making a new function **NonZeroGens**; it'd be easy to make, but would it ever really be useful?

##### #2 - 07 Sep 2023 11:44 - John Abbott

Here is another idea. We could create a constant empty vector of RingElem, and use that.

```
const std::vector<RingElem> NoGens;  
...  
ideal(P, NoGens); // since NoGens is empty it works for all rings P
```

This is probably good for readability, but it does create a new global object.

It is also similar to what one must do in CoCoA-5: namely **ideal(P,[])**

##### #3 - 03 Oct 2023 17:05 - John Abbott

In comment 1 I used **EmptyList**; in comment 2 I used **NoGens**

I think I dislike EmptyList because it is rather ambiguous: it should be **EmptyListOfRingElem** (but who wants to write/read that?)  
**NoGens** is short, but reads well only in a context where "generators" are needed. Might we need an empty list of RingElem in any other context?

**#4 - 04 Jan 2024 10:23 - John Abbott**

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

I still think that ideal(P) looks odd, and is less immediate to understand than ideal(P,EmptyListOfRingElem).  
Or we could just create a new function ZeroIdeal(P).  
What do you think?

**#5 - 22 Jan 2024 11:01 - Anna Maria Bigatti**

Anna: read it all, decide, implement, close!

**#6 - 06 Feb 2024 08:49 - Anna Maria Bigatti**

- Related to Design #1647: Suppress zero from ideal generators? Detect 1 and simplify generators? added

**#7 - 06 Feb 2024 08:50 - Anna Maria Bigatti**

- Subject changed from implement ideal(P) for zero ideal, with no generators? to implement ideal(R) for zero ideal, with no generators?

**#8 - 16 Feb 2024 18:00 - Anna Maria Bigatti**

- Status changed from In Progress to Rejected

- % Done changed from 10 to 100

No longer needed because ideal(zero(R)) now gives ideal in R with no generators.

**#9 - 16 Feb 2024 18:01 - Anna Maria Bigatti**

- Description updated

**#10 - 16 Feb 2024 18:01 - Anna Maria Bigatti**

- Estimated time set to 1.01 h