

CoCoALib - Design #908

Sum of ideals: what are the generators of $(x) + (0)$?

14 Jul 2016 20:44 - Anna Maria Bigatti

Status:	Closed	Start date:	14 Jul 2016
Priority:	High	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	100%
Category:	Improving	Estimated time:	2.01 hours
Target version:	CoCoALib-0.99850	Spent time:	1.70 hour
Description			
The current strategy is that the generators of the sum is the concatenation of the generators of the two ideals. Is this a //rule//? In other words: (x) + (0) should be (x,0) or can it be (x)? (x) + (1) should be (x,1) or can it be (1)? 2024-01 decision "yes"			
Related issues:			
Related to CoCoALib - Design #1647: Suppress zero from ideal generators? Det...		Closed	20 Jan 2022

History

#1 - 02 Sep 2016 14:25 - John Abbott

- Target version changed from CoCoA-5.2.0 spring 2017 to CoCoA-5.2.2

Since we shall need to discuss this, I think it is better to postpone the "delivery date" rather than take hasty action.

My current feeling is that it is OK to do some cleaning of the list of generators.

Related question: if L is a list of polynomials, do we guarantee that $\text{gens}(\text{ideal}(L))$ is equal to L? Or might the result be a "cleaned" version of L? e.g. with zero elements removed, with duplicated removed, perhaps rescaled conveniently.

NOTE even if we do not "clean" the elements of L, might they be ordered differently in $\text{gens}(\text{ideal}(L))$?

#2 - 15 Nov 2017 16:54 - John Abbott

- Target version changed from CoCoA-5.2.2 to CoCoA-5.2.4

I have spoken briefly to Renzo about this.

He thinks it is most natural if $\text{gens}(\text{ideal}(L)) = L$ always.

Ideal sum usually just concatenates the lists of gens, but does handle $\text{ideal}(0)$ specially sometimes:

```
Z := zero(R);
ideal(x) + ideal(Z); --> ideal(x)
ideal(x) + ideal(Z,Z); --> ideal(x)
ideal(x, Z, y) + ideal(Z, y, x, Z); --> ideal(x, 0, y, 0, y, x, 0)
ideal(Z) + ideal(x); --> ideal(0, x)
```

It seems that ideal sum recognizes if the second arg is the zero ideal.

#3 - 20 Nov 2017 14:06 - Anna Maria Bigatti

- Estimated time set to 8.00 h

I have met this problem in a particular context (hyperplane arrangements).
We constructed an ideal and then asked SyzOfGens.
One of the generators was 0, and we did need those syzygies (and their correct length)
[I cannot find it now!! I wrote this so I do not forget...]

#4 - 25 Jul 2018 15:36 - John Abbott

- Target version changed from CoCoA-5.2.4 to CoCoA-5.3.0

#5 - 01 Oct 2019 14:29 - John Abbott

- Target version changed from CoCoA-5.3.0 to CoCoA-5.4.0

#6 - 03 Feb 2022 19:19 - John Abbott

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

#7 - 03 Feb 2022 19:19 - John Abbott

- Target version changed from CoCoA-5.4.0 to CoCoA-5.4.2

#8 - 04 Jan 2024 09:49 - John Abbott

- % Done changed from 10 to 20

Currently I think we **should not** require that $\text{gens}(I+J) = \text{concat}(\text{gens}(I), \text{gens}(J))$. Other ideal operations do not give guarantees about the generators of results (e.g. in $I*J$). If the user really wants an ideal whose generators are $\text{concat}(\text{gens}(I), \text{gens}(J))$ then this can be done explicitly as $\text{ideal}(\text{concat}(\text{gens}(I), \text{gens}(J)))$

I can see that for functions such as SyzOfGens we cannot "clean up" the generators: i.e. we have the rule $L = \text{gens}(\text{ideal}(L))$

There is a function MinSubsetOfGens which will do some cleaning; also we could just do $I + \text{ideal}(R, 0)$

#9 - 12 Jan 2024 10:41 - Anna Maria Bigatti

- Project changed from CoCoA-5 to CoCoALib

- Category changed from enhancing/improving to Improving

- Target version changed from CoCoA-5.4.2 to CoCoALib-0.99850

#10 - 22 Jan 2024 10:44 - John Abbott

Status on 2024-01-22

```
/**/ I := ideal([x,0]);  
/**/ gens(I);  
[x, 0]  
/**/ J := ideal([0,y,z,0]);  
/**/ gens(J);  
[0, y, z, 0]  
/**/ K := I+J;  
/**/ gens(K);  
[x, 0, 0, y, z, 0]
```

I get the same result even if I compute RGBs for I and J before computing the sum.

#11 - 22 Jan 2024 10:47 - John Abbott

Anna has just confirmed that the documentation **does not guarantee** that $\text{gens}(I+J) = \text{concat}(\dots)$. This is good!

#12 - 22 Jan 2024 10:53 - Anna Maria Bigatti

Neither the manual for CoCoALib nor for CoCoA-5 mention that the generators are the mere concatenation, so we are allowed to be "clever". We can remove 0s, deal with 1s, maybe with invertibles, and with MinGens (if pre-computed).

#13 - 22 Jan 2024 10:53 - Anna Maria Bigatti

- *Description updated*

#14 - 22 Jan 2024 10:55 - Anna Maria Bigatti

- *% Done changed from 20 to 50*

#15 - 07 Mar 2024 21:39 - John Abbott

- *Assignee set to Anna Maria Bigatti*

Fully resolved now? Close?

#16 - 08 Mar 2024 14:34 - Anna Maria Bigatti

- *Status changed from In Progress to Closed*

- *% Done changed from 50 to 100*

- *Estimated time changed from 8.00 h to 2.01 h*

Closing because duplicated into [#1647](#)