

## CoCoALib - Design #1649

### Add file SparsePolyOps-vector.C

21 Jan 2022 13:06 - Anna Maria Bigatti

<b>Status:</b>	Closed	<b>Start date:</b>	21 Jan 2022
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	Tidying	<b>Estimated time:</b>	5.01 hours
<b>Target version:</b>	CoCoALib-0.99850	<b>Spent time:</b>	4.55 hours
<b>Description</b>			
Add file SparsePolyOps-vector.C with functions interreduced (now in SparsePolyOps-interreduced.C) and syz(vector) (new)			
<b>Related issues:</b>			
Related to CoCoALib - Bug #1641: gcd does not recognize univariate input		<b>Closed</b>	<b>20 Dec 2021</b>
Related to CoCoALib - Feature #1488: BuiltIn Interreduce-Function		<b>Closed</b>	<b>15 Sep 2020</b>
Related to CoCoALib - Design #1720: DivAlg in CoCoALib		<b>New</b>	<b>21 Dec 2022</b>
Related to CoCoALib - Design #1414: Make class RingElemVector?		<b>In Progress</b>	<b>12 Feb 2020</b>

### History

#### #1 - 21 Jan 2022 13:07 - Anna Maria Bigatti

- Related to Bug #1641: gcd does not recognize univariate input added

#### #2 - 28 Jan 2022 08:23 - Anna Maria Bigatti

I think we should call this SparsePolyOps-VectorRingElem.C, or SparsePolyOps-RingElems.C (too subtle, I fear)

add here:

syz  
DivAlg

move here:

NR  
reduce  
IndetsProd  
IndetsIn  
(taking a std::vector<RingElem>)

and also:

CoefficientsWRT (returning vector)?  
CoeffVecWRTSupport ..  
(returning a std::vector<RingElem>)

With all these, it still remains a reasonably small file.

#### #3 - 28 Jan 2022 08:30 - Anna Maria Bigatti

- Category set to Tidying
- % Done changed from 0 to 20

In SparsePolyOps-RingElem we also have:

FindReducerIndex  
ReduceActiveLM  
reduce

Should these go in SparsePolyOps-VectorRingElem or SparsePolyOps-reduce?

#### #4 - 28 Jan 2022 08:37 - Anna Maria Bigatti

- Category deleted (Tidying)
- % Done changed from 20 to 0

I think we should call this SparsePolyOps-VectorRingElem.C, or SparsePolyOps-RingElem.C (too subtle, I fear)

add here:

~~syz~~ ----> no: must stay in submodule  
DivAlg v ----> to be improved (see Nicolas Jagersma code using GPoly)

move here:

NR v  
reduce v  
IndetsProd v  
IndetsIn v  
(taking a std::vector<RingElem>)

and also:

CoefficientsWRT (returning vector<CoeffPP>) v  
CoeffVecWRSupport .. (returning a std::vector<RingElem>) v

With all these, it still remains a reasonably small file.

Anna Maria Bigatti wrote:

In SparsePolyOps-RingElem we also have:

FindReducerIndex  
ReduceActiveLM  
reduce

Should these go in SparsePolyOps-VectorRingElem or SparsePolyOps-reduce?

(talking to myself)

At the moment SparsePolyOps-reduce contains reduce-related operations, but quite technical/internal (using GPoly and Reductors types).

Now my preference is to keep these 3 functions for "public use" together with the vector-functions. (i.e. "vector<RingElem>" wins over "reduce-related")

**#5 - 28 Jan 2022 09:29 - Anna Maria Bigatti**

Anna Maria Bigatti wrote:

syz

even though this was the start of this topic, Syz of a vector should go in submodule.C, because returns a module.

**#6 - 28 Jan 2022 10:22 - John Abbott**

- *Category set to Tidying*
- *Status changed from New to In Progress*
- *% Done changed from 0 to 10*

It seems that Anna accidentally reset the progress for this issue... maybe she had two browser tabs on the same issue?

I vote **against** SparsePolyOps-ringelems.C: just 1 letter difference is too little!

It is not clear to me where syz should go.

No doubt we have other functions which take inputs of one type and return a result of another.

On the whole it seems more natural to me for the functions to be groups according to the type of parameter that they expect. If we had better documentation for CoCoALib then this would be less of an issue: *e.g.* the user could ask for "all functions returning module".

**#7 - 14 Feb 2022 17:04 - Anna Maria Bigatti**

- *% Done changed from 10 to 60*

Checked in.

Changed version number to 719 (very close to the official 720)

Now the file contains the division related functions.

**#8 - 14 Feb 2022 17:13 - Anna Maria Bigatti**

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

I don't want to make hasty decision now, so this ("division related") is the state for release 0.99720.

Postponing other decisions for next version.

**#9 - 14 Feb 2022 18:18 - John Abbott**

- *Related to Feature #1488: BuiltIn Interreduce-Function added*

**#10 - 07 Mar 2022 10:42 - Anna Maria Bigatti**

We decided syz to be in submodule.C. Checked in.

**#11 - 21 Dec 2022 18:28 - Anna Maria Bigatti**

- Related to Design #1720: DivAlg in CoCoALib added

**#12 - 22 Mar 2024 09:35 - John Abbott**

- Status changed from In Progress to Feedback

- % Done changed from 60 to 90

**#13 - 22 Mar 2024 09:38 - Anna Maria Bigatti**

- Status changed from Feedback to Resolved

Add syz and syz0 (allowing 0s) for vector of RingElem

**#14 - 22 Mar 2024 09:40 - Anna Maria Bigatti**

- Related to Feature #1206: syz, SyzOfGens: which shifts for zero? added

**#15 - 25 Mar 2024 07:42 - Anna Maria Bigatti**

- Status changed from Resolved to Closed

- % Done changed from 90 to 100

Anna Maria Bigatti wrote:

Add syz and syz0 (allowing 0s) for vector of RingElem

**syz** should stay in submodule.C, non in SparsePolyOps-vector.

Closing this issue: the file has been added and already hosts some functions.

**#16 - 25 Mar 2024 08:12 - Anna Maria Bigatti**

- Status changed from Closed to Resolved

- % Done changed from 100 to 90

After closing I realized that the "coefficients" functions were missing and that the .H files were not aligned. Still working on this.

**#17 - 25 Mar 2024 08:22 - Anna Maria Bigatti**

After trying to rearrange the header files I found this problem.

There are functions on vectors of RingElem need headers of RingElem functions, and viceversa. But I find it disturbing to have the cross include of the two H files in both.

I suggest having all headers in SparsePolyOps-RingElem.H (well indicated) and the "vector functions" in the dedicated SparsePolyOps-vector.C, just for convenience.

**#18 - 25 Mar 2024 09:06 - Anna Maria Bigatti**

Anna Maria Bigatti wrote:

I suggest having all headers in SparsePolyOps-RingElem.H (well indicated) and the "vector functions" in the dedicated SparsePolyOps-vector.C, just for convenience.

I tried, and I'm satisfied: some includes are necessary only for the "vector-functions" so the separation of the C files looks meaningful, not just "convenience". On the other hand, one single H file is convenient for practical usage.

**#19 - 25 Mar 2024 09:42 - John Abbott**

- Related to Design #1414: Make class RingElemVector? added

**#20 - 25 Mar 2024 17:49 - Anna Maria Bigatti**

- Status changed from Resolved to Closed

- % Done changed from 90 to 100

**#21 - 25 Mar 2024 17:49 - Anna Maria Bigatti**

- Estimated time set to 5.01 h

**#22 - 26 Mar 2024 09:29 - Anna Maria Bigatti**

- Related to deleted (Feature #1206: syz, SyzOfGens: which shifts for zero?)