

CoCoALib - Feature #1667

GBasis over ZZ: port to CoCoALib

16 Feb 2022 20:00 - John Abbott

<b>Status:</b>	In Progress	<b>Start date:</b>	16 Feb 2022
<b>Priority:</b>	Urgent	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	10%
<b>Category:</b>	Improving	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CoCoALib-0.99880	<b>Spent time:</b>	4.65 hours
<b>Description</b> Port the GBasis over ZZ code (prototype in a CoCoA-5 package) into C++.  Also make several algorithmic improvements. Maybe Florian can indicate some useful texts (at least one by Christian Eder).			
<b>Related issues:</b> Related to CoCoA-5 - Feature #1272: Groebner Bases over ZZ <b>Closed</b> <b>18 Apr 2019</b> Related to CoCoA-5 - Support #242: CoCoA-5 Projects for students (e.g. credit... <b>In Progress</b> <b>28 Sep 2012</b> Related to CoCoALib - Feature #1001: CoCoALib: ideas for student projects <b>In Progress</b> <b>19 Jan 2017</b>			

History

#1 - 16 Feb 2022 20:00 - John Abbott

- Related to Feature #1272: Groebner Bases over ZZ added

#2 - 16 Feb 2022 20:32 - John Abbott

- Related to Support #242: CoCoA-5 Projects for students (e.g. crediti F and tesi) added

#3 - 20 Apr 2022 17:11 - John Abbott

- Related to Feature #1001: CoCoALib: ideas for student projects added

#4 - 26 Sep 2022 12:05 - John Abbott

- Status changed from New to In Progress  
- % Done changed from 0 to 10

A student (Khalil Loukhnati) worked on this last semester. He has produced working code.  
It still needs considerable polishing.

Performance is disappointing (but the profiler reported that 50% CPU time was spent doing downcasting!!)

#5 - 07 Oct 2022 11:34 - Anna Maria Bigatti

This student translated the code in CoCoA prototype-GBZZ.cpkg5 into cocoalib.  
And also added some changed found in a paper by Christian Eder (arxiv:1811.05736 **warning** contains several typos)

Elisa and Michele: want to run some tests and contribute to make it usable?

John Abbott wrote:

A student (Khalil Loukhnati) worked on this last semester. He has produced working code.  
It still needs considerable polishing.

Performance is disappointing (but the profiler reported that 50% CPU time was spent doing downcasting!!)

**#6 - 16 Feb 2024 09:39 - John Abbott**

- Target version changed from *CoCoALib-0.99850* to *CoCoALib-0.99880*

**#7 - 26 Mar 2024 17:19 - John Abbott**

- Priority changed from *Normal* to *Urgent*

Michele Toriell is asking for this. There was C++ from when i was in Passau: dig it out, and blow the dust off.

**#8 - 29 Apr 2024 21:13 - John Abbott**

It seems that we have several implementations: some in C++, and some in CoCoA-5.

The C++ versions ought to be quicker (but perhaps not by much if arithmetic costs dominate).

I would like to compare Loukhnati's impl with that by the student Anna mentioned in comment 5.

One important point is that if a constant element of the ideal is found, it can be used to reduce all coeffs of all other polys (might need to be careful if the LC reduces to zero?). Loukhnati's code produced over 700 constants in one ideal: the smallest was only 5 digits, while other had many thousands of digits...

**#9 - 09 May 2024 20:46 - John Abbott**

The CoCoA-5 version is in *\*prototype-GBZZ.cpkg5\**.

I tried just one test, and the CoCoA-5 code was noticeably faster (maybe 100x). Must investigate!