

CoCoALib - Feature #253

W.Brun's wish list

04 Oct 2012 12:50 - John Abbott

Status:	Closed	Start date:	04 Oct 2012
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Various	Estimated time:	2.49 hours
Target version:	CoCoALib-0.99850	Spent time:	2.40 hours
Description			
Dear Anna and John,			
let me collect all the points that came up in the development of nmzIntegrate (Christof has everything and can show it to you):			
1) Input of polynomials. We have made such a function for nmzIntegrate, but is not good for general use.			
2) A default constructor for RingElem. Not absolutely necessary, but makes programming easier.			
3) Coefficients and power products should automatically be considered as polynomials for arithmetic. This is too natural for every mathematician.			
4) Spaces separating terms of polynomials?			
5) A simpler solution for using OpenMP			
6) Nice conversion from coefficients of polynomials to integers and rationals that can be used outside CoCoA (for example in Normaliz).			
7) More liberal treatment of external libraries by avoiding hardwired paths			
8) Function for homogeneous components (nmzIntegrate contains a function for standard grading)			
9) Standard grading of polynomial rings by default			
10) At one point you ask whether one should use the term "multiplicity" in connection with factorization. I think this is the right choice. I think you are using "exponent" at present.			
11) One could think of a clean up function for polynomials in the following sense: suppose I build a polynomial by freely using PushBack without paying attention to the monomial order. Then I could run such a function to sort the terms and convert my wild list into a correct polynomial.			
Related issues:			
Related to CoCoALib - Feature #221: Better RingElems		Closed	08 Aug 2012
Related to CoCoALib - Support #234: Update/Improve geobucket documentation		Closed	25 Sep 2012
Related to CoCoALib - Feature #223: Automatic mapping of RingElems		Closed	08 Aug 2012
Related to CoCoALib - Feature #222: Printing polynomials - spaces between terms		In Progress	08 Aug 2012
Related to CoCoALib - Feature #209: ReadExpr: input polynomials in CoCoALib		Closed	24 Jul 2012
Related to CoCoALib - Feature #233: AsINT and AsRAT -- ConvertTo<BigInt> and...		Closed	25 Sep 2012
Related to CoCoALib - Feature #224: Leading form		Closed	03 Sep 2012
Related to CoCoA - Feature #247: Use gmp default location if possible		Closed	17 Sep 2012
Related to CoCoALib - Support #256: Improve doc about ordering/grading for po...		Closed	08 Oct 2012
Related to CoCoALib - Feature #342: Remove denominators: QQ[x] -> ZZ[x] (an...		Closed	17 Apr 2013
Related to CoCoALib - Feature #407: RingElem ctor from mpz_t (from Bruns)		Closed	12 Oct 2013
Related to CoCoA - Support #425: Osnabrueck 2014-01		Closed	27 Jan 2014

History

#1 - 04 Oct 2012 12:54 - John Abbott

Implemented Bruns(10): "multiplicity" instead of "exponents" in factorization

#2 - 04 Oct 2012 18:21 - Anna Maria Bigatti

- Subject changed from Bruns's wish list to W.Brun's wish list

#3 - 04 Oct 2012 18:49 - Anna Maria Bigatti

- Status changed from New to In Progress

#4 - 05 Oct 2012 12:21 - John Abbott

There is a new configuration option `--threadsafe-hack` which activates the CoCoALib compilation flag `-DCoCoA_THREADSAFE_HACK`.

As the name suggests it is only a hack, and should not be relied upon to produce robust, leak-free code. We do hope eventually to make a proper threadsafe version of CoCoALib.

This resolves Bruns's point (5).

#5 - 05 Oct 2012 12:25 - John Abbott

I have changed the name of a field in a factorization object: it is now `myMultiplicities` rather than the previous `myExponents`. An analogous change has been made to the record produced by factorization functions in CoCoA-5: the field is now called `Multiplicities` rather than `Exponents`.

This resolves Bruns's point (10).

#6 - 08 Oct 2012 16:25 - John Abbott

Closed issue [#247](#) has satisfied wish (7).

#7 - 17 May 2015 10:27 - John Abbott

Can we finish this issue in the near future?

#8 - 28 Apr 2017 11:28 - Anna Maria Bigatti

- Project changed from CoCoA to CoCoALib

- Target version set to CoCoALib-1.0

This issue was under "CoCoA" instead of "CoCoALib".

I'm recovering these old and forgotten issues, so we reconsider them.

#9 - 28 Apr 2017 15:03 - Winfried Bruns

An issue I stumbled on recently is 3) from my wish list. I think it is not yet implemented. Suppose I produce a polynomial from a monomial M (with coefficient) by substituting polynomials for the indeterminates in the monomial and then I want to multiply this new polynomial by the coefficient of M
...

Also 6) can cause headache.

#10 - 02 May 2017 14:55 - John Abbott

- % Done changed from 0 to 50

Task (6): convert a ringelem to an integer or rational should be fairly easy via **ConvertTo**: here is a simplistic example of the syntax...

```
RingElem x(R);  
...  
BigInt N = ConvertTo<BigInt>(x);  
BigRat q = ConvertTo<BigRat>(x);
```

Task (3): automatic (homomorphic) mapping of ring elems is high on my list of desirable features, but may take a long time to do correctly. Currently the approach I use is the following:

```
PolyRing P = ...;  
RingHom coeff = CoeffEmbeddinghom(P);  
RingElem f(P);  
...  
f = f/coeff(lc(f)); // make f monic
```

In words: I create a RingHom (with a helpful name) and then apply it explicitly where it is needed. I know this is far from ideal, but I think it is not too onerous as a workaround.

#11 - 22 Oct 2020 17:18 - John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99850
- % Done changed from 50 to 70

#12 - 09 Mar 2023 21:57 - John Abbott

- Category set to Various
- Status changed from In Progress to Resolved
- Assignee set to John Abbott
- % Done changed from 70 to 80

Shall we close this very old issue? Most of its children have been completed.

#13 - 14 Mar 2023 20:09 - John Abbott

- Status changed from Resolved to Closed
- % Done changed from 80 to 100
- Estimated time set to 2.49 h