

CoCoA-5 - Feature #90

Make the new fn CoefficientsWRT available in C5

11 Feb 2012 13:25 - John Abbott

Status:	Closed	Start date:	11 Feb 2012
Priority:	Normal	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	100%
Category:	CoCoA-5 function: new	Estimated time:	7.60 hours
Target version:	CoCoA-5.0.9	Spent time:	11.50 hours
Description			
JAA has recently added CoefficientsWRT, ContentWRT, and a few other fns to CoCoALib. Make these fns available to C5 users too -- D'Ali` will want to use them.			
Related issues:			
Related to CoCoALib - Feature #51: polynomial coefficient extraction w.r.t. v...		Closed	
Related to CoCoALib - Feature #278: add CoeffVecWRT to cocoalib		Closed	28 Nov 2012

History

#1 - 14 Feb 2012 15:54 - Anna Maria Bigatti

- Category set to CoCoA-5 function: new
- Assignee set to Anna Maria Bigatti
- Target version set to CoCoA-5.0.2
- % Done changed from 0 to 50

#2 - 14 Feb 2012 16:37 - Anna Maria Bigatti

- % Done changed from 50 to 80

#3 - 01 Mar 2012 16:23 - John Abbott

The interfaces to ContentWRT and CoefficientsWRT are awful -- they seemed reasonable when I designed them, but then I tried to use them...

I propose adding the following interfaces for the case of a single indet:

```
ContentWRT(f, x);  
CoefficientsWRT(f, x);
```

where f represents a polynomial, and x represents a ringelem which is an indet (and **not** the index of the indet!)

If we accept the above "simple" interface then we should also consider the following extension (which I think should probably replace the existing interface):

```
Content(f, [x, y]);  
Coefficients(f, [x, y]);
```

where f is a poly, and x and y are ringelems whose values are indets.

If we accept any of the proposals above then we should consider analogous changes to the interfaces to the homonymous functions in CoCoALib.

#4 - 02 Mar 2012 16:01 - Anna Maria Bigatti

I think we should keep both interfaces: Co**WRT and Co**WRT
The first is very convenient, but the second one is more direct.
Consider this example

```
Use S := QQ[a,b,c];
F := a*b + b*c;
Use R := QQ[x,y,z];
/*do something with*/ ContentWRT(F, [1]);
```

it would inefficient to write

```
ContentWRT(F, Indet(S,1));
```

when the first thing done by the function is to check if Indet(S,1) is an indeterminate...

#5 - 07 Mar 2012 15:32 - John Abbott

Anna Maria Bigatti wrote:

it would inefficient to write...

Yes, but I believe the run-time inefficiency would be negligible.

Perhaps you are right that we should offer both interfaces? Undoubtedly the interface accepting a RingElem whose value is an indet is the most natural and convenient for interactive use. In contrast the "indet index" interface is much closer to the interface available in CoCoALib (apart from the difference of 1 in index values); then again the difference in index values might be a good reason not to offer the "indet index" interface in C5 -- otherwise "mechanical" translating of C5 into CoCoALib code runs the risk of producing code that *looks right* but has the *wrong* behaviour.

#6 - 21 Mar 2012 16:18 - John Abbott

Here are some arguments in favour of the interface which wants a list of RINGELEM to specify which indets:

- JAA believes it is easier to use and more natural than using indices (and probably also less error prone -- wysiwyg)
- it is resilient under changes to the order of indets in the ring specification (or if new indets are introduced)
- using RINGELEM allows run-time checking that the indets specified do indeed belong to the correct ring (this cannot be done with the indices)

While recognizing the validity of Anna's example favouring the use of indices, JAA think it is better to offer just a single type of interface (at least initially). Remember it is easier/safer to extend an interface than to restrict it.

At the moment it is not clear to JAA what the CoCoALib interface should be if we adopt the RINGELEM interface in CoCoA-5. JAA thinks the naturalness and ease-of-use point becomes much weaker in CoCoALib -- the other points remain valid.

#7 - 22 Mar 2012 13:50 - Anna Maria Bigatti

John Abbott wrote:

At the moment it is not clear to JAA what the CoCoALib interface should be if we adopt the RINGELEM interface in CoCoA-5. JAA thinks the naturalness and ease-of-use point becomes much weaker in CoCoALib -- the other points remain valid.

OK. I'll change the cocoa-5 interface. I'm inclined to leave cocoalib as it is (i.e. vector of indices)

#8 - 22 Mar 2012 15:20 - Anna Maria Bigatti

- Status changed from New to Closed
- % Done changed from 80 to 100
- Estimated time changed from 6.00 h to 7.60 h

#9 - 26 Nov 2012 16:41 - Anna Maria Bigatti

- Status changed from Closed to Resolved
- Target version changed from CoCoA-5.0.2 to CoCoA-5.0.9
- % Done changed from 100 to 90

coefficients($4x^3+2x+1$, x); had been disabled.

Write CoefficientsWRT(F, x); returning the coefficients of F wrt x in RingOf(F) (i.e. the dense representation)

In CoCoALanguage

```
Define CoefficientsWRT2(F,x)
  If Type(F)<>RINGELEM Then F := RingElem(RingOf(x),F); EndIf;
  CWRT := CoefficientsWRT(F,[x]);
  D := Max([deg(M.PP)|M In CWRT]);
  Coeffs := NewList(D+1, zero(RingOf(F)));
  Foreach M In CWRT Do Coeffs[deg(M.PP)+1] := M.Coeff; EndForeach;
  Return Coeffs;
EndDefine;
```

#10 - 26 Nov 2012 19:30 - John Abbott

In CoCoALib **CoefficientsWRT** accepts both a RingElem and a *list of indets* as 2nd arg, and produces the same type of result (namely vector<CoeffPP>).

We could consider other names, *e.g.* CoeffListWRT and CoeffVecWRT?

I do have a mild preference for slightly different names (corresponding to the differing return types).

#11 - 28 Nov 2012 11:31 - Anna Maria Bigatti

- *Status changed from Resolved to Closed*

- *% Done changed from 90 to 100*

I added **CoeffListWRT** to NotBuiltin.cpkg5.

Added and fixed manual for CoeffListWRT, coefficients, CoefficientsWRT