CoCoALib - Feature #1299

New fn ConstantTerm?

29 Jul 2019 14:01 - John Abbott

Status:	Closed	**Start date:**	29 Jul 2019
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	New Function	**Estimated time:**	1.66 hour
Target version:	CoCoALib-0.99710	**Spent time:**	1.50 hour

Description

In some prototype code I would like to use a function which gives me the "constant term" of a polynomial. The result may be 0.

One way is to call CoeffOfTerm(f,1); well, actually CoeffOfTerm(f, one(RingOf(f))).
But this is likely to be needlessly slow.

Is it a good idea to have a dedicated function? If so, what should it be called?

History

#1 - 29 Jul 2019 17:04 - John Abbott

Possible names include **ConstTerm** and **ConstantTerm**. The result should be an element of CoeffRing.

Implementing in CoCoALib should be quick; I do not recall now, but maybe CoCoALib has (almost direct) access to the last term in a poly.
It would be good if the code can avoid traversing the whole poly. If we do have to traverse, note that we should first check whether the "next" pointer is null; if it is not, we go to the next term; if it is null, then we check whether the PP is 1 (no need to check for other PPs being equal to 1).

#2 - 29 Jul 2019 18:20 - Anna Maria Bigatti

- % Done changed from 0 to 10

I think it could be useful.
But I wouldn't call it ConstantTerm, because term in CoCoA usually (always?) is equivalent to "power-product", so it would cause ambiguity.
Call it constant(f)?

#3 - 29 Jul 2019 20:33 - John Abbott

- Status changed from New to In Progress

Other candidates:

- **ConstantPart**
- **ValueAtZero** or **ValueAt0**
- **ConstantCoeff** or **ConstantCoefficient**

Further ideas?

JAA quite likes ConstantCoeff at the moment...

#4 - 09 Jan 2020 12:20 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

#5 - 24 Mar 2020 21:49 - John Abbott

This is a little bit like **HomogCompt(f,0)**. See issue #1439.

#6 - 26 Mar 2020 10:28 - John Abbott

- Assignee set to John Abbott

- % Done changed from 10 to 40

I have a prototype impl, much like the impl for **HomogCompt**.

It is disappointingly slow for larger polynomials:

```
use QQ[x,y,z];
f := 1+x+y+z;
g := f^200; -- takes about 90s, occupies about 400Mbytes
ConstantCoeff(g); -- takes about 0.4s; had hoped it'd be faster
```

UPDATE: (2020-04-19) NumTerms(g) takes about 0.48s; result is 1373701

#7 - 19 Apr 2020 19:26 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 40 to 90

- Estimated time set to 1.66 h

I have now checked in my code... since I was checking CVS anyway.
Also added doc to CoCoAHelp.xml.

#8 - 30 Apr 2020 14:20 - Redmine Admin

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99710

#9 - 30 Apr 2020 14:57 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100