# CoCoALib - Feature #222

## Printing polynomials - spaces between terms

08 Aug 2012 20:31 - John Abbott

| | | | |
|---|---|---|---|
| **Status:** | In Progress | **Start date:** | 08 Aug 2012 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 30% |
| **Category:** | Various | **Estimated time:** | 10.50 hours |
| **Target version:** | CoCoALib-0.99880 | **Spent time:** | 4.70 hours |

**Description**

Bruns points out that spaces are inserted in an asymmetrical manner between terms when printing a polynomial. For instance x+1 is currently printed as x +1 which is ugly.

CoCoA-4 prints out x+1 as x + 1 (*i.e.* a space before and a space after the + sign).

We should also consider what happens when the coeffs are themselves polynomials. How should (x-1)*y+(x+1)*z be printed? As an element of QQ[x,y,z]? As an element of QQ[x][y,z]?

**Related issues:**

| | | |
|---|---|---|
| Related to CoCoALib - Bug #74: printing polynomials | **New** | **22 Dec 2011** |
| Related to CoCoALib - Feature #253: W.Bruns's wish list | **Closed** | **04 Oct 2012** |
| Related to CoCoA-5 - Support #242: CoCoA-5 Projects for students (e.g. credit... | **In Progress** | **28 Sep 2012** |
| Related to CoCoALib - Design #432: Semantics of IsPrintedWithMinus | **In Progress** | **31 Jan 2014** |
| Related to CoCoALib - Feature #1117: Better printing of negative coeffs | **In Progress** | **07 Nov 2017** |
| Related to CoCoALib - Design #1156: Printing for RingElem | **New** | **12 Feb 2018** |

### History

#### #1 - 08 Aug 2012 20:38 - John Abbott

*- Estimated time set to 5.00 h*

Bruns suggests inserting no spaces.

This is the easiest solution. We should try it, and decide how readable polynomials look with this convention (perhaps comparing with CoCoA-4).

We should assemble a small database of polynomials (and their rings!) to use as test cases for assessing how "nice" the printed form is.

#### #2 - 04 Sep 2012 10:47 - John Abbott

We could introduce a flag to say whether to print spaces between summands in polynomials.

The flag could be compile time or run-time. Since great speed is not crucial, there is no real advantage to using a compile-time flag. A run-time flag could even be user settable (perhaps belonging to the GlobalManager?).

**Addendum** removing the space before the + or - sign is very simple (just disable line 529 in SparsePolyRing.C); adding a space after the + sign is simple too (change lines 530 and 552), but adding a space after the - sign is more tricky because the - sign is printed as part of the coefficient. Unfortunately the documentation for IsPrintedWithMinus (in the doc for RingElem) is not as clear as I would like.

#### #3 - 04 Sep 2012 12:08 - Anna Maria Bigatti

John Abbott wrote:

> The flag could be compile time or run-time. Since great speed is not crucial, there is no real advantage to using a compile-time flag. A run-time flag could even be user settable (perhaps belonging to the GlobalManager?).

I agree

> **Addendum** removing the space before the + or - sign is very simple (just disable line 529 in SparsePolyRing.C); adding a space after the + sign is simple too (change lines 530 and 552), but adding a space after the - sign is more tricky because the - sign is printed as part of the coefficient. Unfortunately the documentation for IsPrintedWithMinus (in the doc for RingElem) is not as clear as I would like.

If my memory works well the reason why there is no space after the sign came originally from the fact that "-2" (int) is printed like that.
Then I saw (and I'm still convinced) that it's more compact, while nicely separating the summands in a polynomial).

To add the space after "-" we would probably only need to intercept the printing of negative machine integers and print "- " and the absolute value.

CoCoA-4 printed like this (no space for the first term).
What to do?

```
-3x - 1
```

**#4 - 08 Oct 2012 13:10 - John Abbott**

Christof and John looked at various possible printed forms of **(3-2*x)^5**

```
[a]  -32*x^5+240*x^4-720*x^3+1080*x^2-810*x+243
[b]  -32*x^5 +240*x^4 -720*x^3 +1080*x^2 -810*x +243
[c]  -32*x^5 + 240*x^4 - 720*x^3 + 1080*x^2 - 810*x + 243
[d]  - 32*x^5 + 240*x^4 - 720*x^3 + 1080*x^2 - 810*x + 243
```

We both felt that **[a]** is the hardest to read -- it is too uniform, your eye gets "lost" and does not comprehend the structure.
We both felt that **[b]** is acceptable, but not as pleasant as **[c]**.
We both felt that **[c]** looks nicest.
Format **[d]** is confusing when used to print a list of polynomials such as [x,-y,z].

We also looked at the polynomial **(2*a-x)^5** in the ring QQ[a][x]

```
[aa]  -x^5 +(10*a)*x^4 +(-40*a^2)*x^3 +(80*a^3)*x^2 +(-80*a^4)*x +32*a^5
[bb]  -x^5 + (10*a)*x^4 + (-40*a^2)*x^3 + (80*a^3)*x^2 + (-80*a^4)*x + 32*a^5
[cc]  -x^5 + (10*a)*x^4 - (40*a^2)*x^3 + (80*a^3)*x^2 - (80*a^4)*x + 32*a^5
[dd]  -x^5 + 10*a*x^4 - 40*a^2*x^3 + 80*a^3*x^2 - 80*a^4*x + 32*a^5
```

We thought that **[aa]** is just acceptable.
Format **[bb]** was the one we liked most.
Format **[cc]** seems less clear than **[bb]**.
Format **[dd]** is the "lightest" but disguises the structure.

**ADDENDUM** JAA notices that the final term 32*a^5 was not printed in brackets. Why not?

**#5 - 08 Oct 2012 16:21 - John Abbott**

*- Status changed from New to In Progress*

*- % Done changed from 0 to 10*

**#6 - 09 Oct 2012 14:48 - John Abbott**

JAA proposes the following guideline:

a coefficient is printed between brackets except when:

- the coefficients +1 and -1 are handled specially, or
- the coefficient is an integer (*i.e.* IsInteger gives true), or
- the power product is 1 and the coefficient is rational (*i.e.* IsRational gives true)
- negative integer/rational coefficients are handled specially (*i.e.* not ...+(-c)*x^k)

Here are some examples:

```
[A] x^2 - 1   // any ring
[B] x^2 - 1/4 // any ring, special handling for negative rational
[C] x^2 + (-1/4) // any ring
[D] x^2 - a          // element of QQ[a,x]
[E] x^2 - (1/4)*a    // element of QQ[a,x], special handling for negative rational
[F] x^2 + (-1/4)*a   // element of QQ[a,x]
[G] x^2 + (-a)       // element of QQ[a][x]
[H] x^2 + ((-1/4)*a) // element of QQ[a][x]
[I] x^2 + (-(1/4)*a) // element of QQ[a][x], special handling for negative rational
[J] x^2 - (1/2)*x + 1/16  // any ring, no brackets around 1/16, special handling for -1/2
[K] x^2 + (-1/2)*x + 1/16
[L] x^2 + (-1/2)*x + (1/16)
```

Opinions about **[B]** versus **[C]**?
Opinions about **[E]** versus **[F]**?
Opinions about **[J]** versus **[K]** versus **[L]**?
Any other opinions/suggestions/examples?

**2013-02-18** JAA thinks [C] is ugly. JAA mildly prefers [E] to [F], but incompatibly also thinks that [H] is nicer than [I]. Aesthetically [K] looks nicer than [L], but [L] is more uniform.

**#7 - 31 Jan 2014 20:32 - John Abbott**

*- Category set to Various*

This issue has been sitting idle for a year.  We should decide, and then implement!

**Addendum:** JAA thinks that a leading "minus sign" should probably be handled differently from one between two terms.  Here are the examples to consider: -x + 2 and - x + 2 and x^2 - x + 2.

**Addendum2:** the special handling for "leading minus" would be important for printing out a polynomial whose value happens to be an integer (*e.g.* -1)

**#8 - 01 Feb 2014 10:20 - Winfried Bruns**

I would prefer a symmetric appearance, either no space around the + sign or a blank on bothsides.

But it is a matter of taste and adaptation --- if one has seen the asymmetric apperance long enough one gets used to it.

**#9 - 03 Feb 2014 18:24 - John Abbott**

*- % Done changed from 10 to 20*

In note 6 I unwittingly overlooked some (important?) points: for instance I did not consider compound coefficients in the coeff ring.  Here are some more cases to consider.

```
[AA] x^2 + (-a+1)*x + (-a-1)  in QQ[a][x]   --> the coeffs are "compact"
[BB] x^2 + (-a + 1)*x + (-a - 1)  in QQ[a][x]  -->  coeffs have spaces
[CC] x^2 + (a)*x + (a)         in QQ[a][x]
[DD] x^2 + (a)*x + a           in QQ[a][x]
[EE] x^2 + a*x + a             in QQ[a][x] but looks like it is in QQ[a,x]
[FF] x^2 + (-a)*x + (-a)       in QQ[a][x]
[GG] x^2 - (a)*x - (a)         in QQ[a][x]
[HH] x^2 - a*x - a             in QQ[a][x] but looks like it is in QQ[a,x]
[II] a                 in QQ[a][x] but does not look like deg = 0
[JJ] (a)               in QQ[a][x]
[KK] ((a))             in QQ[a][b][x]
[LL] x^2 - x + ((-a))  in QQ[a][b][x]
[ZZ] x^2 - ((1/4)*a)           in QQ[a][x]
```

It now seems to me that "good aesthetics" and "clear structure" do not always go together.  I think that the "clear structure" approach is likely to be easier to implement.

**#10 - 03 Feb 2014 18:53 - John Abbott**

I notice in SparsePolyRing.C:551 that there is a check via IsPrintAtom.  The documentation says *true iff arg does not need brackets when a num or denom of a fraction*

So how should x^2 - x/a - 1/a (elem of QQ(a)[x]) be printed?

```
[aaa] x^2 + (-1/a)*x + (-1/a)
[bbb] x^2 - (1/a)*x - (1/a)
[ccc] x^2 - (1/a)*x - 1/a
```

I think [bbb] is "nicest", and probably [aaa] is ugliest (though possibly the easiest to understand "at a glance").

**#11 - 01 Apr 2014 17:35 - Anna Maria Bigatti**

*- Target version set to CoCoALib-0.99533 Easter14*

**#12 - 08 Apr 2014 18:35 - John Abbott**

*- Target version changed from CoCoALib-0.99533 Easter14 to CoCoALib-0.99534 Seoul14*

**#13 - 14 Jul 2014 17:55 - John Abbott**

*- Target version changed from CoCoALib-0.99534 Seoul14 to CoCoALib-1.0*

**#14 - 18 May 2015 12:16 - John Abbott**

*- Estimated time changed from 5.00 h to 10.50 h*

This issue has been idle for another year.

**#15 - 07 Nov 2017 12:34 - John Abbott**

*- Related to Feature #1117: Better printing of negative coeffs added*

**#16 - 12 Feb 2018 12:33 - John Abbott**

*- Related to Design #1156: Printing for RingElem added*

**#17 - 22 Oct 2020 16:55 - John Abbott**

Idle for more than 6 years: perhaps because there is no clear answer, and implementation might be tricky :-/

**#18 - 14 Mar 2023 20:00 - John Abbott**

**SOURCE CODE** has moved: now near  **SparsePolyOps-RingElem.C:480**

**#19 - 14 Mar 2023 20:08 - John Abbott**

*- % Done changed from 20 to 30*

I am tempted to make the following change (which I hope is not too difficult):

- the coefficient of the PP 1 is printed out the same way as for any other term in the poly

Currently we try to be "clever" and avoid putting brackets around the coefficient (sometimes).
Several example above illustrate what I mean. In contrast the following shows that we do
sometimes use brackets:

```
/**/ use FF7a ::= ZZ/(7)[a];
/**/ I := ideal(a^2-3);
/**/ K := FF7a/I;
/**/ use P ::= K[x];
/**/ (x-2)^3;
x^3 +x^2 +(-2)*x +(-1)
```

Any objections?

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

*- Target version changed from CoCoALib-1.0 to CoCoALib-0.99880*

**#21 - 14 Mar 2023 21:22 - Anna Maria Bigatti**

**THIS REPLY BASED ON MY MISTAKE IN COMMENT 19**
On my computer I get

```
/**/ FF7a ::= ZZ/(7)[a];
/**/ use P ::= FF7a[x];
/**/ (x-2)^3;
x^3 +x^2 -2*x -1
```

and I much prefer that (without the parentheses).

I think I am missing something in your proposal.
I believe we should close this issue (originated for deciding spaces around signs) and make new ones, more specific.

**#22 - 14 Mar 2023 21:27 - John Abbott**

Sorry I typed in the example wrongly: there should have been a quotient (now corrected -- see comment 19)