

## CoCoALib - Feature #735

### Convert a PPMonoidElem to RingElem with coefficient one

11 Jun 2015 08:31 - Mario Albert

<b>Status:</b>	Closed	<b>Start date:</b>	11 Jun 2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	New Function	<b>Estimated time:</b>	2.00 hours
<b>Target version:</b>	CoCoALib-0.99540 Feb 2016	<b>Spent time:</b>	1.90 hour
<b>Description</b>			
Let pp be a PPMonoidElem of the PPMonoid PPM. Let P be a SparsePolyRing with PPM as PPMonoid. Several times I want to handle pp as a member of P. At the moment I have to call for this case			
<pre>RingElem r = monomial(P, One(CoeffRing(P)), pp); // r = 1*pp</pre>			
It would be quite nice if there would be a method monomial(SparsePolyRing, PPMonidElem) which can do the same:			
<pre>RingElem r = monomial(P, pp); // r = 1*pp</pre>			

#### History

##### #1 - 11 Jun 2015 08:41 - Anna Maria Bigatti

- Category set to New Function
- Status changed from New to In Progress
- Target version set to CoCoALib-0.99540 Feb 2016
- Estimated time set to 2.00 h

That seems very reasonable: I had a look at the code and, indeed, most of the times monomial is called the coefficient is 1. I can see no ambiguity, so we can do it.

Meanwhile you may use the more convenient call monomial(P,1,pp) ;-)

##### #2 - 11 Jun 2015 08:56 - Anna Maria Bigatti

Should we do the same for monomial(P, expv)?  
I cannot see any ambiguity.

**REPLY** (JAA) if we have monomial(P,PP) then we should also have monomial(P,expv)

##### #3 - 11 Jun 2015 10:41 - John Abbott

- % Done changed from 0 to 10

I am a little uncertain about the proposal. monomial(P,pp) gains little over monomial(P,1,pp).

Is it worth creating a "new" function for relatively little benefit?

Do you think that monomial(P,pp) is usefully clearer to read than monomial(P,1,pp)?

I think it is marginally clearer, but am uncertain whether it really merits a "new" function.

Perhaps the documentation should make it clearer that `monomial(P,1,pp)` works as expected, and that it is not necessary to write `monomial(P,one(CoeffRing(P)),pp)`.

I do not believe that `monomial(P,1,pp)` is measurably slower than `monomial(P,one(...),pp)`; and even if it were slower, we could treat the case of `coeff=1` specially to make it faster.

I'm not opposing the proposal; I just want to be convinced that it is worthwhile. Perhaps better documentation is the right approach...?

#### #4 - 11 Jun 2015 10:46 - Mario Albert

I didn't know that `monomial(P, 1, pp)` works as expected.

But I think for the *normal* user who wants to use a `PPMonoidElem` as a `RingElem` it would be more natural if he can use `monomial(P, pp)` as well.

#### #5 - 11 Jun 2015 11:00 - John Abbott

- % Done changed from 10 to 20

I do agree that, without familiarity with the function, upon reading `monomial(P,1,pp)` my first thought would be "What does 1 mean here?" In contrast, `monomial(P,pp)` does seem clearer.

I note that `monomial(P,pp)` could also be written `RingElem(P,pp)`, but the former seems to express better what the programmer wants (the latter is perhaps too vague).

Is there any other name we should consider? Probably it should have the same name as the 3 arg version...

#### #6 - 11 Jun 2015 11:04 - John Abbott

I point out that assignment to `RingElem` can perform some type changes. Should we allow assignment to a `RingElem` from `PPMonoidElem`? This would succeed only if the `RingElem` belongs to a poly ring `P` and the `PPMonoidElem` belongs to `PPM(P)`.

I'm less convinced about this idea (implicit conversion) than Mario's original proposal (explicit conversion).

#### #7 - 11 Jun 2015 11:24 - Anna Maria Bigatti

My preference (by far) goes to `monomial(P, pp)`  
I have already implemented and all tests pass.  
Should I also do `monomial(PP, expv)`? I think so.

#### #8 - 11 Jun 2015 11:46 - John Abbott

- Assignee set to Anna Maria Bigatti

OK, do it! Including `monomial(P,expv)`.  
Don't forget documentation, examples and tests!

#### #9 - 11 Jun 2015 19:00 - Anna Maria Bigatti

- Status changed from In Progress to Resolved

- % Done changed from 20 to 70

done, checked in.

So far I modified only the files in AlgebraicCore

**#10 - 20 Oct 2015 14:55 - John Abbott**

Hi Anna. Could you finish this issue, and take it to "feedback"? Thanks, J

**#11 - 22 Mar 2016 17:59 - Anna Maria Bigatti**

- Status changed from Resolved to Closed

- % Done changed from 70 to 100

updated calls in BuiltinFunctions-CoCoALib and CoCoALibSupplement.

NOTE: this is only done for SparsePolyRing because monomial(R, i) would be ambiguous for DensePolyRing.

Closed.