

CoCoALib - Design #582

Error codes: use same code for "not poly ring" and "not elem poly ring"

07 Jul 2014 09:25 - John Abbott

Status:	New	Start date:	07 Jul 2014
Priority:	Normal	Due date:	
Assignee:		% Done:	30%
Category:	Tidying	Estimated time:	3.00 hours
Target version:	CoCoALib-0.99880	Spent time:	1.00 hour
Description			
Proposal: merge the two error codes NotPolyRing and NotElemPolyRing.			
Reasons: (1) the distinction is really quite subtle (2) it makes some of the source code simpler (3) it reduces the number of error codes			
Obviously also applies to other similar pairs of error codes.			
Related issues:			
Related to CoCoALib - Feature #385: Design new errors using inheritance		In Progress	08 Jul 2013
Related to CoCoALib - Feature #92: Error Codes		In Progress	14 Feb 2012
Related to CoCoALib - Design #427: Error names and error messages (current de...		In Progress	28 Jan 2014
Related to CoCoALib - Feature #1793: Use ErrorContext instead of string FnName		In Progress	16 Mar 2024

History

#1 - 07 Jul 2014 09:29 - John Abbott

See also emails exchanged on 20140704 with subject "Opinion".

We need suitable wording for the combined error: how about these?

- (A) Expected (element of) PolyRing
- (B) Operation valid only for PolyRing

Further ideas?

#2 - 07 Jul 2014 09:36 - Anna Maria Bigatti

John Abbott wrote:

- (A) Expected (element of) PolyRing

this one!

#3 - 15 Jul 2014 15:08 - John Abbott

Just not to lose it in a mountain of archived emails, here is the decisive message:

```
J>> My "enthusiasm" for merging 'ExpectedPolyRing' and "ExpectedElemPolyRing" comes from
J>> several reasons:
J>> (1) it make some of my code simpler (see below)
J>> (2) it reduces the number of error codes (perhaps won't matter when we reorganize them?)
J>> (3) there is very little conceptual difference between them
```

```
A> I agree with all 3!
A>the conceptual difference is so small it does not justify making the code more tedious to write.
A> The balance goes for one error only.
A> Convinced!
A> (I'm checking in the tests)
```

Here's the code I mentioned:

CURRENT CODE

```
long NumTerms(ConstRefRingElem f)
{
    if (!IsPolyRing(owner(f)))
        CoCoA_ERROR(ERR::NotElemPolyRing, "NumTerms(f)");
    return PolyRingPtr(owner(f)->myNumTerms(raw(f)));
}
```

POTENTIAL NEW VERSION

```
long NumTerms(ConstRefRingElem f)
{
    return PolyRingPtr(owner(f), "NumTerms(f)")->myNumTerms(raw(f));
}
```

Unfortunately the presence of the fn name string makes the line a bit less readable.

Perhaps the tricky part would be wording the error message so that makes sense to a normal use in both circumstances...

What do you think?

#4 - 17 Jul 2014 14:35 - John Abbott

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

#5 - 19 Apr 2017 09:29 - Anna Maria Bigatti

- Related to Design #427: Error names and error messages (current design) added

#6 - 26 Mar 2024 14:36 - Anna Maria Bigatti

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

- % Done changed from 0 to 30

I think this has been agreed upon a long time ago, as explained in [#582-3](#).

Moreover the recent addition of the context makes it quite easy ([#1793](#)).

I tested it on the NumTerm function:

```
long NumTerms(ConstRefRingElem f)
{ return PolyRingPtr(owner(f), CoCoA_ERROR_CONTEXT)->myNumTerms(raw(f)); }
```

and in CoCoA-5 I get:

```
/**/ NumTerms(zero(QQ));
--> ERROR: Ring must be a polynomial ring
--> [CoCoALib] NumTerms
--> NumTerms(zero(QQ));
--> ^^^^^^^^^^^^^^^^^^^^^
```

#7 - 26 Mar 2024 14:37 - Anna Maria Bigatti

- Related to Feature #1793: Use ErrorContext instead of string FnName added