CoCoALib - Bug #736

QuotientRing: is it correct to prohibit quotient by ideal(1)?

21 Jun 2015 21:06 - John Abbott

Status:	Closed	**Start date:**	21 Jun 2015
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	Improving	**Estimated time:**	1.11 hour
Target version:	CoCoALib-0.99700	**Spent time:**	1.05 hour

Description

The quotient ring R/I where I=ideal(1) gives the trivial ring.

Currently the quotienting operation gives a specific error because we do not want users trying to compute in the trivial ring. However this also blocks reasonable operations such as dim(R/I) and multiplicity(R/I).

Should we really forbid the creation of the trivial ring?

History

#1 - 21 Jun 2015 21:09 - John Abbott

The problem arose while doing some computations for Ulrich where it was known/expected that some ideals would be ideal(1).

Is there an easy way to test whether an ideal is ideal(1)? We have IsZero, but not IsOne.

#2 - 25 Jun 2015 12:52 - John Abbott

Robbiano confirms that it is best to avoid the null ring (which has surprising properties such as 1 = 0).

In this case we should offer an easy/natural way of testing whether an ideal is ideal(1). What do you think about the readability of the following:

- if 1 isin I then ....
- if IsOne(I) then ....

Given that IsZero already exists, it seems coherent to offer IsOne too (for which types of arg?)

#3 - 25 Jun 2015 14:30 - Anna Maria Bigatti

- Status changed from New to In Progress

- Assignee set to Anna Maria Bigatti

- % Done changed from 0 to 10

John Abbott wrote:

- if 1 isin I then ....
- if IsOne(I) then ....

the first already works. The second should be easy to add (I can do it).

#4 - 25 Jun 2015 15:34 - Anna Maria Bigatti

- Status changed from In Progress to Resolved

- % Done changed from 10 to 30

done: added IsOne(IDEAL)

#5 - 27 Jan 2020 17:43 - John Abbott

- Status changed from Resolved to Closed

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99700

- % Done changed from 30 to 100

- Estimated time set to 1.11 h

This was presumably done about 4 years ago, but we forgot to close the issue...
Closing now.