CoCoA-5 - Feature #877

Easier syntax to make a PRINCIPAL ideal?

09 May 2016 11:11 - John Abbott

Status:	Closed	**Start date:**	09 May 2016
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	enhancing/improving	**Estimated time:**	0.77 hour
Target version:	CoCoA-5.2.0 spring 2017	**Spent time:**	0.75 hour

Description

Currently the way to make a principal ideal is not so natural: the generator must be put into a list.

Are there any objections to allowing the natural syntax ideal(x) where x is a RingElem?

History

#1 - 09 May 2016 11:43 - Anna Maria Bigatti

It is already done, both in CoCoA-5 and CoCoALib (in CoCoALib up to 4 generators).

#2 - 09 May 2016 14:51 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Now I recall what I tried: ideal(R,x) does not work; it says it wants a list instead of a RINGELEM.

Are there any objections to allowing also ideal(RING, value)?
If the value is a RINGELEM then specifying the RING is redundant (but not harmful).
We could allow value to be of type INT (or RAT?); in which case ideal(R, value) would presumably be equivalent to ideal(RingElem(R, value)).
Mmmm, now I see it written down it does not seem to offer much gain over inserting explicitly the call to RingElem.

We could still activate a new signature ideal(RING, RINGELEM); or else perhaps change the error message? I did feel a bit of a fool when demo-ing CoCoA-5 that it seemed one has to write ideal(R, [x]) instead of ideal(R,x)... of course, I should have remembered that ideal(x) works.

Perhaps the error message should say which signatures ideal does offer?

#3 - 10 May 2016 14:02 - Anna Maria Bigatti

- % Done changed from 10 to 0

John Abbott wrote:

> Perhaps the error message should say which signatures ideal does offer?

Yes, I like that (I have already done something similar for len(M[1]), for example)

#4 - 17 Jun 2016 15:19 - Anna Maria Bigatti

- Related to Feature #357: Constructor for vectors? CoCoAVector added

#5 - 17 Jun 2016 15:19 - Anna Maria Bigatti

- Related to deleted (Feature #357: Constructor for vectors? CoCoAVector)

#6 - 10 Oct 2016 17:05 - Anna Maria Bigatti

- Status changed from In Progress to Feedback

- % Done changed from 0 to 90

Close this issue? I think that ideal(x) is hard to beat ;-)

#7 - 14 Oct 2016 14:31 - John Abbott

The source code the **ideal** function in CoCoA-5 appears to start at about line 1520 in BuiltInFunctions-CoCoALib.C.

What does DECLARE_ARITYCHECK_FUNCTION do? Why is the first check in DECLARE_BUILTIN_FUNCTION a check on the arity?

#8 - 14 Oct 2016 14:43 - John Abbott

Getting a better error message from the input ideal(R,x) would be decidedly tricky. The error is actually triggered by evalArgAsListOfRingElem which does not know who called it.

Probably we should abandon hopes of getting a better error message (at least, for the time being). Anyway, the manual page is quite clear...

I think we can close this issue (but I am still curious about the true function of the ARITYCHECK function).

#9 - 27 Apr 2017 14:57 - John Abbott

- Status changed from Feedback to Closed

- Assignee set to John Abbott

- % Done changed from 90 to 100

- Estimated time set to 0.77 h

Closing after 7 months in feedback.